Hierarchical clustering of bipartite data sets based on the statistical significance of coincidences

Ignacio Tamarit$^{1,2}$, María Pereda$^{1,2,3}$, and José A. Cuesta$^{1,2,4,5,*}$

$^1$Grupo Interdisciplinar de Sistemas Complejos (GISC), Departamento de Matemáticas de la Universidad Carlos III de Madrid, Leganés, Spain
$^2$Unidad Mixta Interdisciplinar de Comportamiento y Complejidad Social (UMICCS), Madrid, Spain
$^3$Grupo Grupo de Investigación Ingeniería de Organización y Logística (IOL), DIOADE, Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, Madrid, Spain
$^4$Instituto de Biocomputación y Física de Sistemas Complejos (BIFI), Universidad de Zaragoza, Zaragoza, Spain
$^5$UC3M-Santander Big Data Institute (IBiDat), Getafe, Spain

*Corresponding author: cuesta@math.uc3m.es

Abstract

When a set ‘entities’ are related by the ‘features’ they share they are amenable to a bipartite network representation. Plant-pollinator ecological communities, co-authorship of scientific papers, customers and purchases, or answers in a poll, are but a few examples. Analysing clustering of such entities in the network is a useful tool with applications in many fields, like internet technology, recommender systems, or detection of diseases. The algorithms most widely applied to find clusters in bipartite networks are variants of modularity optimisation. Here we provide an hierarchical clustering algorithm based on a dissimilarity between entities that quantifies the probability that the features shared by two entities is due to mere chance. The algorithm performance is $O(n^2)$ when applied to a set of $n$ entities, and its outcome is a dendrogram exhibiting the connections of those entities. Through the introduction of a ‘susceptibility’ measure we can provide an ‘optimal’ choice for the clustering as well as quantify its quality. The dendrogram reveals further useful structural information though—like the existence of sub-clusters within clusters. We illustrate the algorithm by applying it first to a set of synthetic networks, and then to a selection of examples. We also illustrate how to transform our algorithm into a valid alternative for uni-modal networks as well, and show that it performs at least as well as the standard, modularity-based algorithms—with a higher numerical performance. We provide an implementation of the algorithm in Python freely accessible from GitHub.
1 Introduction

Among the networks that we can find in real life, bipartite networks stand on their own because of their special nature. Bipartite (two-mode) networks divide their nodes into two different categories, and links join nodes of one category only with nodes of the other. Bipartite networks can be used to describe plant-pollinator mutualistic interactions [1, 2, 3], words in documents [4, 5], scientists and co-authored papers [6, 7], genes in viral genomes [8, 9], actors in films [10, 11], people attending events [12], recommender systems [13], etc., and they have been successfully applied to problems ranging from internet technology [14, 15] to systems biology and medicine [16]. A defining feature of any system amenable to bipartite-network modelling is that one set can be thought of as ‘entities’ and the other one as ‘features’. For instance, if the entities are scientists, the features are papers they author—or vice versa, if the entities are the papers, the features are their authors. Which is the set of entities and which the set of features very much depends on the problem one aims to solve, because a typical question regarding this kind of datasets is: how do entities cluster according to their set of features?

Finding clusters (also called modules or communities) in networks has been an active topic of research for a few decades (see [17] and references therein). There is no clear-cut definition of what a cluster or community is. Intuitively, one expects that nodes in a cluster are more densely connected to each other than to the nodes outside the cluster, but the actual definition is part of the answer to the clustering problem. For this reason, there are a plethora of different methods to determine clusters, and although there is some overlapping in their outcomes, they hardly obtain exactly the same partition of the set of nodes. Which method to choose is then a problem-dependent issue.

Bipartite networks require a purposely definition, because of their very particular connectivity—nodes of the same type are never connected to each other. Roughly speaking, three kinds of approaches have been explored. The most direct one amounts to projecting the network on the type of nodes whose clustering is sought [18, 19, 20]. The result is a weighted network linking these nodes and only these—to which standard clustering algorithms can be applied. The success of this approach very much relies on a suitable choice of the weights for the links.

The second approach is, so to speak, global in nature. A typical method amounts to defining a function of the partition of nodes (‘modularity’), and then finding the partition that maximises it [21]. This function compares the actual linking of the network with that a random null model would produce. The clustering problem then becomes the problem of finding the partition that maximises the modularity of the network. Extending this method to bipartite networks requires choosing a suitable null model [22, 23]. An alternative to modularity is to adopt a Bayesian viewpoint by introducing a stochastic block model [13] whose parameters are obtained through likelihood maximisation [24]. Although the use of these global methods to determine the community structure of a network is widespread, their application to large datasets is limited because they are computationally demanding (they boil down to performing a combinatorial optimisation). Furthermore, the very definition of modularity has some resolution limitations that preclude these methods from detecting clusters that are particularly small [25].

The last approach to the problem is represented by a set of methods that go under the common name of hierarchical clustering [26, ch. 4]. The idea of hierarchical clustering is to define a ‘dissimilarity’ (often a true mathematical ‘distance’) between entities based on the features that they do or do not share, and then sequentially merge the least
dissimilar clusters (initially every node is a cluster), following some prescription. The outcome of these methods is not a partition, but a dendrogram, i.e., a rooted tree in which nodes are grouped according to the dissimilarity value at which they merged into the same cluster. They look very much like phylogenetic trees and can be interpreted similarly. If needed, one can obtain a partition out of a dendrogram either by introducing a dissimilarity threshold or by detecting groups of branches that separate very near the root. As a matter of fact, the seminal work on community detection in networks uses a particular form of hierarchical clustering [27].

There are two main reasons why there is a current preference for global methods over hierarchical clustering. One is the fact that on the latter the definition of clusters eventually depends on the choice of an arbitrary threshold—or a similar ad hoc criterion. The other is the vast amount of different dissimilarity measures that people have used in the literature [28]—each one yielding a different result [26, ch. 3]. Nevertheless, the upside of these methods is that they can be computationally more efficient because they do not involve any combinatorial optimisation process. If \( n \) denotes the number of entities, hierarchical clustering algorithms exist with time complexity \( O(n^2) \) [29, 30].

As of the fact that the result of hierarchical clustering is a dendrogram, from which clusters need to be defined ad hoc, this can be an advantage rather than a drawback. Dendrograms provide a sort of multi-resolution clustering where one can see not only the main clusters, but also sets of nodes forming clusters within clusters—something that may be very informative for some applications (hence the success of phylogenetic trees in evolutionary biology).

The true disadvantage of hierarchical clustering compared to methods based on modularity or stochastic blocks is not only that choosing the right dissimilarity measure is a problem, but that none of these measures uses a null model to decide whether the dissimilarity found between two entities may be spurious [28, 26]—as global methods do. To illustrate the problem, consider the case of words (the entities) in documents (the features). Suppose further that the subject of these documents is ‘politics’. It is clear that a word like ‘politician’ is likely to appear in many of them; but on the other hand, words like prepositions appear in every single document, so the dissimilarity between, say, the word ‘of’ and the word ‘politician’ will be low regardless of the measure we have chosen. And yet, this low dissimilarity is spurious because there is no meaningful connection between these two words. This is the reason why some datasets require an ad hoc pre-processing before one of these algorithms can be applied to the data (for instance, in the processing of texts, it is common to remove words bearing no actual meaning, like articles, prepositions, etc.).

The purpose of this paper is to introduce a random null model for bipartite networks and define a dissimilarity measure between pairs of entities in terms of the statistical significance of the shared and unshared features. This will automatically remove spurious relationships such as the one just described. Combined with, e.g., SLINK, an efficient algorithm for single-linkage clustering [29], it will lead to an \( O(n^2) \) algorithm to generate a dendrogram from bipartite datasets. Additionally, as we shall see, the algorithm can be readily extended to the case of one-mode networks.

2 Description of the method and the algorithm

Consider a set of entities \( \mathcal{E} \) and a set of features \( \mathcal{F} \) with \( |\mathcal{E}| = N_E \) and \( |\mathcal{F}| = N_F \) elements respectively. Each entity will have some of these features, so a bipartite network can be
defined with nodes $E \cup F$ and (bidirectional) links joining entities with their features. The adjacency matrix of such a network has the form

$$A = \begin{pmatrix} 0 & B \\ B^T & 0 \end{pmatrix}, \quad (1)$$

where $B = (b_{ir})$, $i \in E$, $r \in F$, is such that $b_{ir} = 1$ if entity $i$ has feature $r$ and $b_{ir} = 0$ otherwise. Accordingly,

$$n_{ij} = (BB^T)_{ij}, \quad m_{rs} = (B^TB)_{rs}, \quad (2)$$

count the number of features that entities $i$ and $j$ have in common, and the number of entities having both features $r$ and $s$, respectively. In particular, $n_i = n_{ii}$ counts the number of features of entity $i$ and $m_r = m_{rr}$ counts the number of entities having feature $r$.

With these numbers one can introduce all kinds of dissimilarity measures [26, 28] with which to construct a hierarchical clustering and produce a dendrogram for the entities revealing which of them are closer to each other. Instead, we will compute what is the probability that entities $i$ and $j$ have at least $n_{ij}$ features in common if features are assigned randomly to entities (without any bias).

The probability distribution $p(n_{ij}|n_i, n_j, N_F)$ is obtained as follows. We tag all the $n_i$ elements of set $F$ that correspond to features of entity $i$, and then draw, randomly and without replacement, $n_j$ features out of the set $F$. The sought probability is the probability that exactly $n_{ij}$ of these extracted elements are tagged, and it is given by the hypergeometric distribution [31, §2.6]

$$p(n_{ij}|n_i, n_j, N_F) = \frac{\binom{n_i}{n_{ij}} \binom{N_F-n_i}{n_j-n_{ij}}}{\binom{N_F}{n_j}}. \quad (3)$$

What we are interested in is the $p$-value

$$p_{ij} = \sum_{k \geq n_{ij}} p(k|n_i, n_j, N_F). \quad (4)$$

This will be our measure of dissimilarity between entities $i$ and $j$.

Interestingly, this is a very standard problem in statistics that can be solved building the contingency table

|               | drawn    | not drawn | total   |
|---------------|----------|-----------|---------|
| shared feature| $n_{ij}$ | $n_i - n_{ij}$ | $n_i$   |
| not shared feature | $n_{j} - n_{ij}$ | $N_F + n_{ij} - n_i - n_j$ | $N_F - n_i$ |
| total         | $n_j$    | $N_F - n_j$ | $N_F$   |

and applying Fisher’s exact test (FET), for which very efficient algorithms are implemented in widely used programming languages such as Python (scipy.stats.fisher_exact) and R (fisher.test). The outcome of this test is precisely $p_{ij}$, which allows us to easily build the dissimilarity matrix $D = (p_{ij})$. Given $D$, we can apply any agglomerative clustering method using the Lance-Williams algorithm, parametrized as single linkage, for updating the dissimilarity between clusters [26, ch. 4] and get the desired dendrogram representing the clustering structure of the entities.
Once we obtain a dendrogram, deciding the optimal number of clusters (if any) might not be a trivial matter. Several cluster validity indexes (CVIs) have been proposed as a way of selecting the best number of clusters, usually based on between- and within-cluster distances [32]—in a metric, mathematical sense. Given that our dissimilarity matrix $D$ is not a matrix of true distances (in the strict sense), we propose a different approach. In particular, we will choose the partition that maximizes the susceptibility of the system as used in percolation theory. This susceptibility is formally defined as $\chi = \sum n_s s^2 / N$, where $n_s$ is the number of clusters of size $s$, and the sum is taken over all but the largest cluster (see [33] for more details). Notice that the maximum possible value of $\chi$ is achieved when the network breaks into two equally sized clusters, yielding $\chi_{\text{max}} = N/4$. To make this measure independent of the network size—hence comparable—we use $\chi = 4 \sum n_s s^2$ as our normalised susceptibility. For the sake of consistency, we will use $\chi$ to select the optimal number of clusters in all the case studies reported in this manuscript. This notwithstanding, we would like to recall that the multi-resolution nature of hierarchical clustering still provides useful information, and that the actual best partition of particular data is a problem-dependent question.

3 Data Analysis

We test the performance of our algorithm with different datasets. Firstly, we generate synthetic networks with a well-defined community structure and challenge the algorithm to uncover it. Secondly, we use real-world data from congressional voting records (U.S.A) and try to classify the Congressmen in their corresponding political parties—which act as background truth for the underlying community structure. Lastly, we use data from a massive survey carried out in France in 2003 to analyze how some leisure activities are more related than others.

3.1 Computer-Generated Networks

We begin by analyzing bipartite networks created synthetically with a community structure established ex ante. All of these networks will consist of 100 entities and 400 features, connected following different heuristics so that we have a reliable background truth with which to compare the results provided by the algorithm.

The most extreme case of a bipartite network with community structure is a network created as the union of two separate (bipartite) ones. To build such a network, we first select 50 entities and 200 features and create a link between any two of them with probability $1/2$. Then, we take the remaining nodes (50 entities and 200 features) and proceed similarly. When these two networks are put together, the result is, by construction, a two-cluster bipartite network. In Fig. 1A we can see how the algorithm captures this situation seamlessly. The entities are grouped into two different clusters (red and black), which exactly correspond to the original building blocks of the network. Furthermore, the (normalised) susceptibility peaks at $p$-value $p = 1$, where the two clusters split, and achieves its maximum possible value, $\chi = 1$—see SI Fig. S1. Notice that some weak structure can be observed within the two main clusters. It is just caused by random fluctuations [34], and the low values of $\chi$ (SI Fig. S1) confirm this fact.

We now apply the algorithm to a purely random network. We generate it from the previous two-cluster network by adding links with probability $1/2$ between the entities
Figure 1: Analysis of synthetic bipartite networks. (A) dendrogram of a two-cluster bipartite network. (B) dendrogram for an intermediate case with $p_{add} = 0.28$. (C) dendrogram of a random network. In (A), (B), and (C) the dashed lines mark the point with highest susceptibility—that where the ‘optimal’ partition should be found.

of each block and the features of the other. The identity of the two blocks has thus disappeared, and any pattern observed should be spurious. The resulting dendrogram is depicted in Fig. 1C. Just a glimpse to this figure reveals that there is no clear structure—something that the low values of the susceptibility ($\chi = 0.134$) at the threshold point ($p = 0.013$) confirm (SI Fig. S2). The high $p$-value at the threshold also confirms that the
Figure 2: Analysis of synthetic bipartite networks. Maximum value of the normalized susceptibility (which suggests the point of optimal partition) as a function of $p_{\text{add}}$, the probability that a link connects with a node at the ‘wrong’ module. For each value of $p_{\text{add}}$ we have generated 100 realisations of the networks. The values of the susceptibility are the averages over these realisations and the error bars indicate the corresponding standard deviations.

Clustering has low statistical significance.

These results show that the algorithm performs well for the two extreme cases that we have devised. But we can also test it for intermediate cases. We generate these intermediate benchmarks by connecting nodes of opposite modules with varying probability $0 < p_{\text{add}} < 1/2$ ($p_{\text{add}} = 0$ would correspond to the two-block network and $p_{\text{add}} = 1/2$ to the random network). Figure 1B shows what the clusters formed looks like for an intermediate case $p_{\text{add}} = 0.28$. The identity of a few nodes can no longer be recovered, but the two cluster are still clearly identifiable. Figure 2 shows the largest value of the susceptibility for several values of $p_{\text{add}}$, which identifies the point of the ‘optimal’ partition for each network. The susceptibility remains close to 1 up to $p_{\text{add}} \approx 0.25$, indicating that the two modules are clearly identified even if one fourth of the links connect to the ‘wrong’ module. Then the susceptibility undergoes a sharp decay and beyond $p_{\text{add}} \approx 0.35$ it practically vanishes—as for the random network. Notice that Fig. 1B illustrates a case within this region of sharp decay of the susceptibility. What we see in this figure is representative of what happens—the identity of the two clusters gets degrading as $p_{\text{add}}$ increases.

3.2 Congressional Voting Records

The U.S. Congressional voting records dataset [35] gathers all roll call votes made by the United States Congress during the years 1789-2017. Each Congressman is represented by a set of features describing how he voted on every bill for a Chamber (Senate or House), for a particular Congress (period of two years). There are nine different ways of voting; the so-called ‘cast codes’ (see ref. [35] for more details). Prior to analyzing the data, we process them as in [36], that is, we group cast codes 1, 2, and 3 (ways of voting ‘yea’), cast codes 4, 5, and 6 (ways of voting ‘nay’), and cast codes 0, 7, 8, and 9 (not voting). As a result, we build bipartite networks which consist of Congressmen (entities) linked to their particular vote (‘yea’, ‘nay’, or ‘not voting’) on a particular bill (features).

It is well-known that Congressmen usually vote according to their political parties commandments—more so in the recent period [37]. Hence, our algorithm should be able to determine the political party of the different Congressmen based on how they voted the different bills. To test this hypothesis, we analyze the Congressional voting records of both the House and Senate for Congresses 114th (years 2015-2016) and 36th (years
1959-1960).

The results for both Congresses confirm our hypothesis. The algorithm detects two main clusters, and these clusters correspond to the different political parties (Democrats and Republicans). In Fig. 3 we present results for Senate 114, in which Congressmen are more polarised (exhibiting higher values of susceptibility, SI Fig. S12) and the number of Congressmen is smaller than those in House’s datasets—hence making it more suitable for visual interpretation of the dendrogram. We predict Congressmen groups (membership to party) with a 92% of accuracy for Senate 114, 98.41% for House 114 (SI Figs. S6 and S7), 88.57% for Senate 36 (SI Figs. S8 and S9), and 78.37% for House 36 (SI Figs. S10 and S11). As a matter of fact, this lower accuracy in the results for the 36th Congress is not attributable to a lower performance of the algorithm, but to the higher polarisation trend that Congressmen have undergone over the years [37]. The effect is also observed in other clustering techniques (see SI Figs. S13-S16 for a multiple correspondence analysis [38] of the same data).

![Dendrogram of the clustering of roll cast votes of Congressmen for Senate 114. Colours red and black in the dendrogram identify clusters of Congressmen; grey colour is used for Congressmen not assigned to any cluster. Labels in horizontal axis identify Congressmen and their colour identifies the political party they belong to. (The dendrogram is cut at p-values around 10^{-300} because below this value the computation yielded underflows.)](image)

**3.3 “Life story” dataset**

Data from surveys can be represented as a bipartite network. Indeed, the respondents can be regarded as the entities and their possible answers as the features, so that a link between them exists if a respondent chooses a particular answer. Hence, data collected via surveys are amenable to be clustered using the algorithm we described in section 2.

We illustrate this application of the method by analysing data from the 2003 INSEE (Institut National de la Statistique et des Études Économiques) survey on identity construction, the so-called “life story” survey [39]. The study was conducted in the metropolitan areas of France and recorded answers are related to family and professional situation, geographic and social origins, ethical commitments, cultural practices, and state of health. In particular, the dataset we analyse comprises the answers of 8403 people (55% female)
to the question ‘Which of the following leisure activities do you practice regularly?’, and
the answer choices were: Reading, Listening to music, Cinema, Shows, Exhibitions, Com-
puter, Sport, Walking, Travel, Playing a musical instrument, Collecting, Voluntary work,
Home improvement, Gardening, Knitting, Cooking, Fishing, Number of hours of TV per
day on average (0-4). In addition to this information, the data includes four supplemen-
tary variables: sex, age, profession, and marital status (see SI for a brief summary of
descriptive statistics). The dataset is available within the R package FactoMineR [40].

With these data, we build a bipartite network of people (entities) and leisure activities
(features) and apply the algorithm to find groups of activities whose co-appearance in the
individuals’ answers is statistically significant. Each of the activities is represented by two
nodes in the set of features, one corresponding to practicing it, and the other to not doing
so. That way, we can get clusters that include doing some activities and not doing some
other ones. The dendrogram represented in Fig. 4 exhibits two main clusters: the black
one groups all the nodes corresponding to actively performing activities, whereas the red
one collects all the nodes corresponding to not performing them. On the other hand,
watching TV does not seem to be particularly related to any of these clusters. Therefore,
the main contrast between the different leisure activities is, precisely, whether they are
actively performed or not.

These two clusters are identified by the peak of the susceptibility. However, each of
them is further structured in interesting subgroups. For instance, watching movies at the
cinema and spending time with the computer appear as two very closely related activities.
Moving one step above in the hierarchical dendrogram we see that these two activities are
also commonly performed by people who enjoy attending to shows and, moving even one
step further, by people who enjoy exhibitions and traveling.

More relevant information is gained if we include the variables male (sex = 0) and
female (sex = 1) as features. Figure. 5 shows the resulting dendrogram. We can observe
that the activity more closely related to sex is knitting, which seems to be predominately
practiced by females. Not only that: a bit below the first splitting into two main clusters
(at the peak of the susceptibility) there is a secondary splitting that associates fishing,
gardening, mechanics, not cooking, and not knitting with males (and the opposite with fe-
males), whereas activities such as walking, reading, listening to music, practicing sports,
going to shows, exhibitions or the cinema, using computers, and travelling, form a cluster
weakly related to sex. Lastly, the analysis also reveals that activities such as collecting or
volunteering are not preferred by any particular sex.
Figure 4: Dendrogram of the clustering of leisure activities in the “life story” dataset. The dashed line marks the $p$-value with the largest susceptibility.

Figure 5: Same as Fig. 4, but including the supplementary variable ‘sex’.

3.3.1 Comparison with multiple correspondence analysis

For the “life story” data set there is no absolute background truth that we can use to validate our results. However, we can compare them with what is obtained applying one of the most commonly used techniques for analysing the structure of categorical data: multiple correspondence analysis (MCA), an adaptation of standard correspondence analysis to categorical data [38]. Like principal component analysis, MCA represents the data as points in a low-dimensional Euclidean space that retains the maximum variance of the data. A Chi-square test is used to examine whether rows and columns of a contingency table are statistically significantly associated, and component analysis decomposes the chi-squared statistic associated with this table into orthogonal factors (dimensions). Eventually, MCA can be used to find groups of categories (features) or individuals (entities) that are similar.

We permormed MCA on the “life story” data set using the R package FactoMineR [40]. In the so-called factors map (see Fig. 6), the distance between two features is a measure of their similarity. Each feature is represented at the barycenter of the individuals in it.
Features with a similar profile are grouped together, and negatively correlated features appear on opposite sides of the plot origin (opposed quadrants). The distance between feature points and the origin measures the quality of the feature points on the factor map. Feature points that are away from the origin are well represented on the factor map. In our example, this representation only captures a 24% of the variance of the data set.

![Factor map of the “life story” data set. In green, the activity is performed; in black, the activity is not performed.](image)

Grouping features in a MCA factor map is rather subjective. In Fig. 6 we have represented our data points using two different colors: green for actively performed activities, and black for non-performed ones. The points seem to be cluttered in two groups, separated by an imaginary diagonal going from the second to the fourth quadrant. This separation is congruent with the two main clusters detected by our algorithm (see Fig. 4), signalling that this division is indeed present in the data (except for the fact that watching TV is grouped within the green cluster, whereas the dendrogram of Fig. 4 shows that it is an activity that has no special attachment to any group). However, looking for further structure using the factor map is rather complicated. Let us recall that a factor map is a two-dimensional representation of the data that only captures part of its variance (24% in this case). Hence, the distance between any two points in the map can not be trusted as a measure of their true similarity.

Unlike MCA, the clustering algorithm we propose does not lose information by projecting into a lower dimensional space, and takes into account the fact that co-occurrences between features can be due to chance (unlike other hierarchical methods [28, 26]). Hence, even though both techniques are coincident in their coarse grain classification of the data, the method we introduce here is able to unveil finer substructure that remains hidden in the MCA map, and can provide deeper insights when analysing survey data—or any other data with bipartite structure.
4 Application to one-mode networks

As a final application of our algorithm, we will illustrate how it can be applied to ordinary, one-mode networks. As we discussed in the introduction, when trying to analyse bipartite networks, a typical procedure is to project them onto one-mode ones, so that one can take advantage of the many techniques developed for this type of graphs [18] (alternatively, these techniques can be sometimes extended to be directly applied to two-mode networks [41]). What is rarely done is the reverse procedure, that is, to create a bipartite network out of a one-mode one in order to benefit from techniques initially tailored for the former. We propose here a simple idea to do this. What we gain from doing so is that we can exploit the faster performance of our algorithm to quickly produce a multi-resolution clustering of an ordinary network.

Consider the network $G$ with a set of nodes $N$ and adjacency matrix $C$—which we consider symmetric with zero diagonal (no self-loops) for simplicity. Now, we identify $N$ with the set of entities, $E$, create a replica of the same set, and identify it with the set of features, $F$. Links joining nodes of $N$ now join nodes of $E$ with its neighbouring nodes in the replica $F$, thus transforming the original network $G$ into a bipartite network. Furthermore, in order to eliminate the possibility for this bipartite network to be disconnected, we need to link each node of $E$ with its own replica in $F$. Therefore, the bipartite network will be described by the adjacency matrix

\[
A = \begin{pmatrix}
0 & I + C \\
I + C & 0
\end{pmatrix},
\]

and accordingly

\[
n_{ij} = \left( (I + C)^2 \right)_{ij} = \delta_{ij} + 2C_{ij} + (C^2)_{ij},
\]

i.e., $n_{ii}$ is one plus the degree of node $i$, and $n_{ij}$, with $i \neq j$, counts the number of common neighbours that nodes $i$ and $j$ have—plus 2 if $i$ and $j$ are themselves neighbours.

4.1 Tests

To illustrate this application of the algorithm, we turn to two well-known benchmarks, one-mode networks with community structure: the Zachary’s karate club study [42, 27], and the College Football dataset [27]. In the first of them, Zachary monitored the relationships of 34 individuals attending a Karate club that eventually split into two different ones. Very often in the literature, the performance of community detection algorithms is assessed by how well they predict this partition [17]. Our algorithm accomplishes this task almost perfectly, classifying incorrectly only two nodes. One of them is node 9, which Zachary himself misclassified in his study [42]. Furthermore, in Fig. 7 we can detect at least one clear subgroup composed by nodes 17, 6, 7, 5, and 11 (all in black), which is also captured by other classic algorithms for one-mode networks [27].

The College Football network is formed by a set of 115 College Football teams (nodes) which are connected to each other if they were confronted during the regular-season of the Division I in 2000 (U.S.A.) [27]. In reality, the different teams are divided into what is known as conferences, each containing between 8 and 12 teams. Intraconference games are more common than interconference ones, so teams belonging to the same conference are highly interconnected in the network, and a community detection algorithm should be able to account for this. In Fig. 8 we can appreciate how our algorithm does so. The labels at the horizontal axis represent the different clubs, colored according to the conferences.
they belong to. As we can see, most of them are grouped together under the same branch in the dendrogram, which is able to uncover the structure of the conferences. Let us note that the branches are coloured according to the partition that maximizes the susceptibility, which should be taken as a guidance, but that, as in this case, it might not correspond to the (actual) best partition (see also the discussion in section 2).

![Figure 7: Dendrogram of Zachary’s Karate Club network. The dashed line correspond to the point of largest susceptibility.](image)

Figure 7: Dendrogram of Zachary’s Karate Club network. The dashed line correspond to the point of largest susceptibility.

![Figure 8: Dendrogram of the American Football dataset. The dashed line correspond to the point of largest susceptibility.](image)

Figure 8: Dendrogram of the American Football dataset. The dashed line correspond to the point of largest susceptibility.

5 Discussion and conclusions

There are lots of algorithms to analyse clustering in networks, bipartite or otherwise, so why another one? The algorithm we presented here has certain important advantages with respect to previous ones. The main one is its performance. It is based on two operations: first of all, performing a FET between every pair of entities, and secondly, running SLINK to obtain a hierarchical clustering of the entities based on the outcomes of the pairwise
Both operations have a complexity $O(n^2)$ when applied to a bipartite network of $n$ entities—or to a one-mode network with $n$ nodes. The fastest modularity-based algorithm requires a matrix diagonalisation, whose complexity is $O(n^3)$.

But this is not the only advantage of our algorithm: its outcome is a multi-resolution analysis of the relations between the nodes in the form of a dendrogram. If so needed, we can make use of the susceptibility measure that we have introduced to determine an ‘optimal’ partition of the nodes. This measure has the bonus of quantifying the quality of the partition (the higher the susceptibility the better the partition). But we should not neglect that the multi-resolution clustering provided by the dendrogram contains useful information that remains hidden in standard clustering algorithms—see for example our analyses of survey data in section 3.3 and how they compare to standard techniques such as MCA (section 3.3.1). Furthermore, the dissimilarity introduced by the FET is statistically meaningful: it measures the probability that observing those coincidences between the features of two entities is purely due to chance.

The availability of a dendrogram prevents some problems inherent to the optimisation of a network measure such as modularity. It has been shown [43] that modularity-based methods can find spurious structure in random networks. Fluctuations are a source of meaningless associations, but as we have shown, they are easy to spot on a dendrogram. To begin with, there are no obvious clusters that partition the network in big blocks, and furthermore, the $p$-values for which associations occur are too high to be statistically meaningful. Of course, the calculation of the susceptibility will always provide an ‘optimal’ partition even for a random network, however its small value is an indication that this clustering is not to be trusted.

Our algorithm is very easy to implement as well, given the availability of efficient algorithms for calculating the $p$-value of a FET and for performing the hierarchical clustering. As a matter of fact, we provide an open-access, documented implementation of the complete algorithm in Python for public download. The code is available on GitHub https://github.com/mpereda/clusterBip.

**Acknowledgements**

This research has been funded by the Spanish Ministerio de Ciencia, Innovación y Universidades-FEDER funds of the European Union support, under project BASIC (PGC2018-098186-B-I00).

**References**

[1] Memmott, J. (1999) The structure of a plant-pollinator food web. *Ecol. Lett.*, 2, 276–280.

[2] Bascompte, J., Jordano, P., Melián, C. J., and Olesen, J. M. (2003) The nested assembly of plant-animal mutualistic networks. *Proc. Natl. Acad. Sci. USA*, 100, 9383–9387.

[3] Dormann, C. F. and Strauss, R. (2014) A method for detecting modules in quantitative bipartite networks. *Methods Ecol. Evol.*, 5, 90–98.
[4] Srivastava, A., Soto, A. J., and Milios, E. (2013) Text Clustering Using One-mode Projection of Document-word Bipartite Graphs. In Proceedings of the 28th Annual ACM Symposium on Applied Computing New York: ACM SAC ’13 pp. 927–932.

[5] Dhillon, I. S. (2001) Co-clustering documents and words using Bipartite Spectral Graph Partitioning. In Proceedings of the Seventh ACM SIGKDD New York: ACM pp. 269–274.

[6] Newman, M. E. J. (2001) Scientific collaboration networks.I. Network construction and fundamental results. Phys. Rev. E, 64, 016131.

[7] Newman, M. E. J. (2001) The structure of scientific collaboration networks. Proc. Natl. Acad. Sci. USA, 98, 404–409.

[8] Iranzo, J., Krupovic, M., and Koonin, E. V. (2016) The Double-Stranded DNA Virosphere as a Modular Hierarchical Network of Gene Sharing. mBio, 7, e00978–16.

[9] Shapiro, J. W. and Putonti, C. (2018) Gene Co-occurrence Networks Reflect Bacteriophage Ecology and Evolution. mBio, 9, e01870–17.

[10] Peixoto, T. P. (2013) Parsimonious Module Inference in Large Networks. Phys. Rev. Lett., 110, 148701.

[11] Peixoto, T. P. (2014) Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. Phys. Rev. X, 4, 011047.

[12] Freeman, L. C. (2003) Finding Social Groups: A Meta-Analysis of the Southern Women Data. In Breiger, R., Carley, K., and Pattison, P., (eds.), Dynamic Social Network Modeling and Analysis, pp. 39–97 The National Academies Press Washington, D.C.

[13] Guimerà, R., Llorente, A., Moro, E., and Sales-Pardo, M. (2012) Predicting Human Preferences Using the Block Structure of Complex Social Networks. PLoS ONE, 7, e44620.

[14] Cai, J. and Liu, W. X. (2013) A New Method of Detecting Network Traffic Anomalies. In Instruments, Measurement, Electronics and Information Engineering Stafa-Zurich: Trans Tech Publications Ltd Vol. 347 of Applied Mechanics and Materials, pp. 912–916.

[15] Xu, K., Wang, F., and Gu, L. (2014) Behavior Analysis of Internet Traffic via Bipartite Graphs and One-Mode Projections. IEEE ACM T. Network., pp. 931–942.

[16] Pavlopoulos, G. A., Kontou, P. I., Pavlopoulou, A., Bouyioukos, C., Markou, E., and Bagos, P. G. (2018) Bipartite graphs in systems biology and medicine: a survey of methods and applications. GigaScience, 7.

[17] Fortunato, S. and Hric, D. (2016) Community detection in networks: A user guide. Phys. Rep., 659, 1–44.

[18] Zhou, T., Ren, J., Medo, M., and Zhang, Y.-C. (2007) Bipartite network projection and personal recommendation. Phys. Rev. E, 76, 046115.
[19] Martinez-Romo, J., Araujo, L., Borge-Holthoefer, J., Arenas, A., Capitán, J. A., and Cuesta, J. A. (2011) Disentangling categorical relationships through a graph of co-occurrences. Phys. Rev. E, 84, 046108.

[20] Tumminello, M., Miccichè, S., Lillo, F., Piilo, J., and Mantegna, R. N. (2011) Statistically Validated Networks in Bipartite Complex Systems. PLoS ONE, 6, e17994.

[21] Newman, M. E. J. and Girvan, M. (2004) Finding and evaluating community structure in networks. Phys. Rev. E, 69, 026113.

[22] Barber, M. J. (2007) Modularity and community detection in bipartite networks. Phys. Rev. E, 76, 066102.

[23] Guimerà, R., Sales-Pardo, M., and Amaral, L. A. N. (2007) Module identification in bipartite and directed networks. Phys. Rev. E, 76, 036102.

[24] Larremore, D. B., Clauset, A., and Jacobs, A. Z. (2014) Efficiently inferring community structure in bipartite networks. Phys. Rev. E, 90, 012805.

[25] Fortunato, S. and Barthélemy, M. (2007) Resolution limit in community detection. Proc. Natl. Acad. Sci. USA, 104, 36–41.

[26] Everitt, B. S., Landau, S., Leese, M., and Stahl, D. (2011) Cluster Analysis, Wiley, West Sussex, UK 5th edition.

[27] Girvan, M. and Newman, M. E. J. (2002) Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA, 12, 7821–7826.

[28] Gower, J. C. and Legendre, P. (1986) Metric and Euclidean Properties of Dissimilarity Coefficients. J. Classif., 3, 5–48.

[29] Sibson, R. (1973) SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method. Comput. J., 16, 30–34.

[30] Defays, D. (1977) An Efficient Algorithm for a Complete Link Method. Comput. J., 20, 364–366.

[31] Feller, W. (1968) An Introduction to Probability Theory and Its Applications, Vol. 1, Wiley, New York 3rd edition.

[32] Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010) Understanding of Internal Clustering Validation Measures. In 2010 IEEE International Conference on Data Mining pp. 911–916.

[33] Murase, Y., Török, J., Jo, H.-H., Kaski, K., and Kertész, J. (Nov, 2014) Multilayer weighted social network model. Phys. Rev. E, 90, 052810.

[34] Guimerà, R., Sales-Pardo, M., and Amaral, L. A. N. (Aug, 2004) Modularity from fluctuations in random graphs and complex networks. Phys. Rev. E, 70, 025101.

[35] Lewis, Jeffrey B., Keith Poole, Howard Rosenthal, Adam Boche, Aaron Rudkin, and Luke Sonnet (2019). Voteview: Congressional Roll-Call Votes Database. https://voteview.com/ Accessed: 2019-06-11.
[36] Porter, M. A., Mucha, P. J., Newman, M. E. J., and Warmbrand, C. M. (2005) A network analysis of committees in the U.S. House of Representatives. *Proc. Natl. Acad. Sci. USA*, **102**(20), 7057–7062.

[37] Andris, C., Lee, D., Hamilton, M. J., Martino, M., Gunning, C. E., and Selden, J. A. (04, 2015) The Rise of Partisanship and Super-Cooperators in the U.S. House of Representatives. *PLoS ONE*, **10**(4), 1–14.

[38] Abdi, H. and Valentin, D. (2007) Multiple correspondence analysis. *Encyclopedia of measurement and statistics*, **2**, 651–66.

[39] Life story survey in 2003. Institut National de la Statistique et des Études Économiques. [https://www.insee.fr/en/metadonnees/source/operation/s1384/presentation](https://www.insee.fr/en/metadonnees/source/operation/s1384/presentation) Accessed: 2019-02-18.

[40] Ll, S., Josse, J., and Husson, F. (2008) FactoMineR: An R Package for Multivariate Analysis. *J. Stat. Soft.*, **25**(1), 1–18.

[41] Everett, M. G. and Borgatti, S. P. (2013) The dual-projection approach for two-mode networks. *Social Networks*, **35**(2), 204–210.

[42] Zachary, W. W. (1977) An information flow model for conflict and fission in small groups. *J. Anthropol. Res.*, **33**(4), 452–473.

[43] Guimerà, R., Sales-Pardo, M., and Amaral, L. A. N. (2004) Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, **70**, 025101.