Clique Graphs and Overlapping Communities

Article *in* Journal of Statistical Mechanics Theory and Experiment · September 2010

1 author:

Tim S Evans
Imperial College London
123 PUBLICATIONS   **2,643** CITATIONS

Some of the authors of this publication are also working on these related projects:

Embedding Graphs in Lorentzian Spacetime View project

Clique Graphs and Overlapping Communities

T.S. Evans

Theoretical Physics, and the Institute for Mathematical Sciences, Imperial College London, SW7 2AZ, U.K.

Abstract.   It is shown how to construct a clique graph in which properties of cliques of a fixed order in a given graph are represented by vertices in a weighted graph. Various definitions and motivations for these weights are given. The detection of communities or clusters is used to illustrate how a clique graph may be exploited. In particular a benchmark network is shown where clique graphs find the overlapping communities accurately while vertex partition methods fail.

1. Introduction

Much of the work on networks is from a vertex centric viewpoint. We talk about distributions of vertex degree, the clustering coefficient of vertices, and vertex partitions as communities. For instance, consider table 1 which shows the frequency of words in a review of networks [1]. If we ignore stop words such as "the" and "a", and use the stems of words (so 'edg' represents "edge", "edges", "edged", etc) then as table 1 shows the second most popular stem after 'network' is 'vertic' followed by 'edg'. Taking synonyms into account reinforces this picture. Further, edges may often be referred to in the context of the calculation of some vertex property, such as degree.

Stem	Rank	Count	Stem	Rank	Count
network	1	254	number	11	58
vertic	2	107	distanc	12	48
edg	3	86	model	13	47
random	3	86	connect	14	46
graph	5	81	data	15	40
degre	6	78	link	16	38
power	7	68	world	16	38
lattic	8	67	hub	33	25
law	9	65	point	38	23
vertex	10	61	site	40	22

Table 1. Table showing the frequencies of the main network related words in the review of networks [1]. In calculating the frequencies, 'stop words' (such as "the") were removed and then remaining words were stemmed (so the stem 'edg' counts both "edge" and "edges"). The rank is by the number of occurrences of each word.

In some cases this focus on vertices is appropriate. Perhaps, on the other hand, this predominance of vertex concepts reflects an inherent bias in the way we humans conceptualise networks. One way to compensate for our vertex centric view of the original network is to represent other structures of a network, here cliques, in terms of the vertices of a new derived graph. We may then exploit our natural bias in the analysis of the new derived graph while at the same time avoiding our propensity for vertices in the original network.

Cliques – complete subgraphs — are an important structure in graph theory. The name originates from representation of cliques of people in social networks [2]. They have since been used for many purposes in social networks [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. Triads, cliques of order three, are of particular interest. One example is the idea that the most important strong ties (in the language of Granoveter [18, 19] need to be defined in terms of their membership of triads [3, 9, 10, 12, 16].

Cliques are also at the centre of some interesting graph theoretical and algorithmic problems. Finding the set of all 'maximal' cliques (a clique is maximal only if it is not a

subgraph of another clique) is a good example for which the Bron-Kerbosch algorithm [20] is the classic solution.

Cliques are often used to analyse the general structure of a network, for example see [21, 22]. A particular application is to use cliques in the search for communities, as in [23, 24, 25] for example, or equivalently what are called cohesive groups in social network analysis [5, 11, 14]. Alternatively they have been used to produce a model of growing networks [26, 27].

Given the importance of cliques we ask if we can shift our focus away from vertices and onto the cliques of our graph of interest, G, by constructing a "clique graph" in which the vertices of the clique graph represent the cliques of the original graph and the way they overlap. Once this has been done, one can use the standard tools to analyse the properties of the vertices of the clique graph in order to derive information about the cliques in the original graph G. There are many such vertex centred tools but to illustrate the principle we shall look at one complex example, that of finding communities in networks, the topic of cohesion in the social networks literature, clustering in the language of data mining.

The vast majority of community detection algorithms produce a partition of the set of vertices [28, 29]. That is each vertex is assigned to one and only one community. These may be appropriate for many examples, such as those used to illustrate or test vertex partition algorithms. However it is an undesirable constraint for networks made of highly overlapping communities, with social networks being an obvious case. There one envisages that the strong ties are formed between friends where there is a high probability of forming triads through different types of relationship [30, 31]. However friendships may be of different types, family relationships, work collaborations, links formed through a common sport or hobby. In this case it makes no sense to try to assign a single community to each individual but it does make sense to hypothesise that each triad can be given a single characterisation, here a single community label. To find such communities, we will construct the clique graph and then apply a good vertex partitioning algorithm to the clique graph. Thus we will illustrate the general central principle of this paper, namely that a clique graph enables one to avoid the bias of a vertex centric world to study networks in terms of their cliques while at the same time exploiting the very same widely available vertex based analysis techniques to do the analysis at no extra cost.

In the next section we will look at why it is important to construct clique graphs with weights and how this may be done. As an example of how to use vertex based measures on a clique graph to study cliques in the original graph, in section 3 we will construct various overlapping communities. Finally in section 4 we will consider how this approach can be generalised and it fits in with clique overlap in the literature.

2. Clique Graphs

2.1. Incidence Graph Projections

Let us consider a simple graph G with vertices drawn from a vertex set $\mathcal{V}$ and which we will label using mid Latin characters, i, j, etc. Now let us consider the set of all possible order n cliques†, $\mathcal{C}^{(n)}$, for a *single* value of $n \geq 2$. That is $\mathcal{C}^{(n)}$ is the set of complete subgraphs of G with n distinct vertices. We will use early Greek letters, α, β, etc to index these order n cliques. For instance in the graph G shown in figure 1 there are three triangles or order three cliques. For $n = 2$ the order two cliques are just the edges of the original graph G.

The relationship between the order n cliques and the vertices of G can be recorded in an order n clique incidence matrix $B_{i\alpha}^{(n)}$. The entries of this $|\mathcal{V}| \times |\mathcal{C}^{(n)}|$ matrix are equal to 1 if clique $\alpha \in \mathcal{C}^{(n)}$ contains vertex $i \in \mathcal{V}$, otherwise it is 0

$$B_{i\alpha}^{(n)} = \begin{cases} 1 & \text{if } i \in \alpha \in \mathcal{C}^{(n)} \\ 0 & \text{if } i \notin \alpha \in \mathcal{C}^{(n)} \end{cases} . \tag{1}$$

It is useful to define the degree of each vertex i in this bipartite graph as $k_i^{(n)}$ where

$$k_i^{(n)} = \sum_\alpha B_{i\alpha}^{(n)} . \tag{2}$$

This is simply the number of order n cliques which contain vertex i so $k_i^{(2)}$ is simply the usual definition of the degree of vertex i. This order n clique incidence matrix of G may be seen as the adjacency matrix of a bipartite network, $B^{(n)}(G)$, where the two types of vertices correspond to the vertices and the order n cliques of the original graph G. This is shown for the example graph in figure 1.

We can construct a new weighted graph $A^{(n)}(G)$ which is a subgraph of the original graph G by defining its adjacency matrix A as follows

$$A_{ij}^{(n)} = \sum_{\alpha \in \mathcal{C}^{(n)}} B_{i\alpha}^{(n)} B_{j\alpha}^{(n)} (1 - \delta_{ij}) , \qquad \forall\, i, j \in \mathcal{V} . \tag{3}$$

This (3) is the projection of the bipartite incidence graph $B^{(n)}(G)$ onto a unipartite graph $A^{(n)}(G)$. The vertex set of $A^{(n)}(G)$ is identical to the original graph G. The weight of edge (i, j) is the number of order n cliques containing that edge. However, $A^{(n)}(G)$ is not in general the same as G as any edge not in an order n clique in the original graph will not be in the $A^{(n)}(G)$ graph. Each vertex has degree $k_i^{(n)}$ which can be less than the degree of the same vertex i in the original graph. In particular any vertex of degree less than n in G will be isolated in $A^{(n)}(G)$, and any edges in G incident to such a vertex, will not appear in $A^{(n)}(G)$. In the example of figure 1, the only difference between $A^{(3)}(G)$ and G are the two edges on the extreme left and right, neither of which are in any triangles.

† These are distinct from what are referred to as 'n-cliques' in the social networks literature which are not complete subgraphs [5, 11]. However sometimes 'n-cliques' has also been used to refer to the order n cliques of interest here, e.g. see [22].
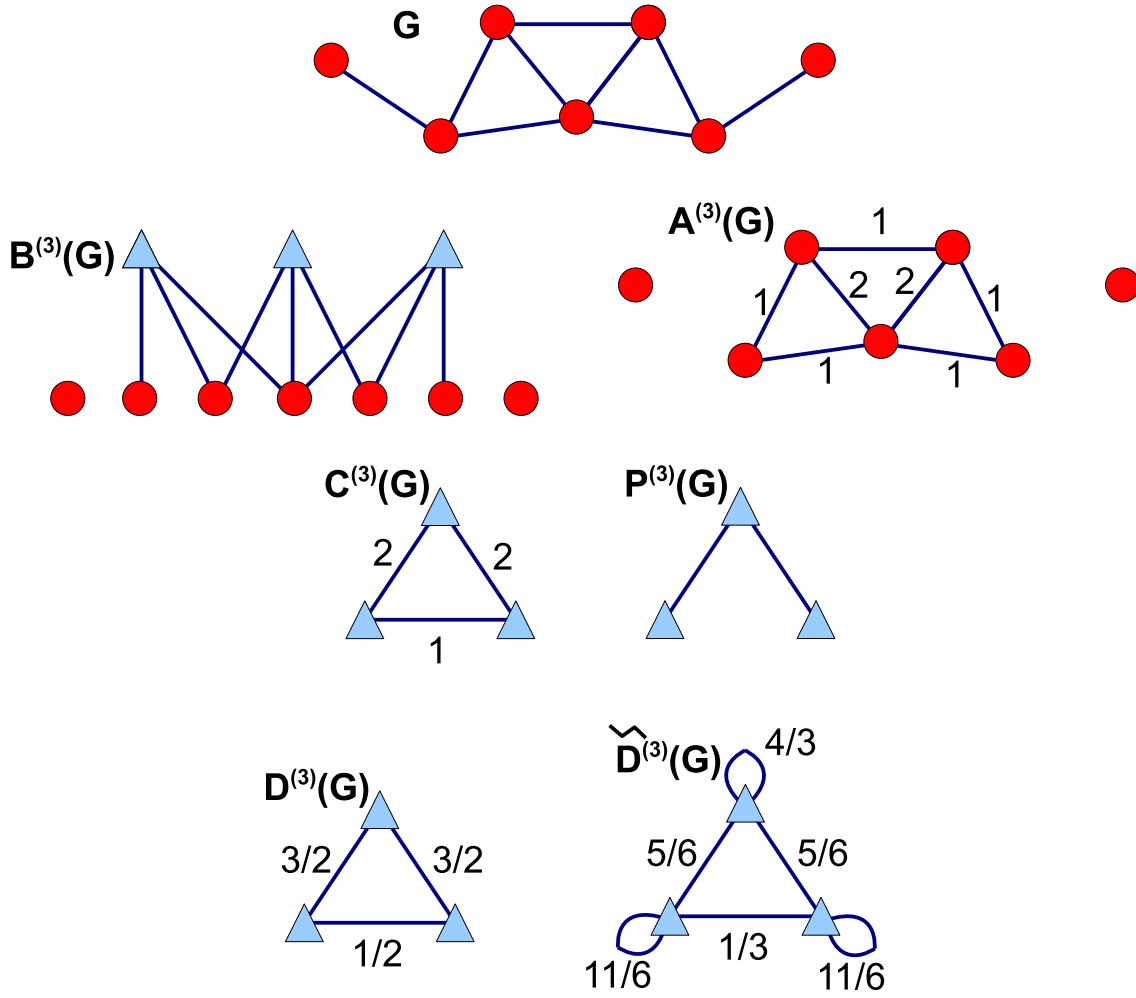
Figure 1. An example of the various graphs defined in this paper for the case of $n = 3$. Here the original graph G shown has three order three cliques, and two vertices in no cliques at all. The order three clique incidence matrix $B^{(3)}(G)$ is a bipartite network whose circle vertices are vertices of G while the three triangle vertices come from the three order three cliques of G. The incidence matrix can be used to define another graph $A^{(3)}(G)$ whose unweighted form is isomorphic to a 3-uniform hypergraph but which is distinct from the original graph G. The clique graphs denoted $C^{(3)}(G)$, $D^{(3)}(G)$ and $\tilde{D}^{(3)}(G)$ correspond to the adjacency matrices defined in (4), (6) and (7) respectively. The unweighted versions of these clique graphs are identical to the standard line graph of the 3-uniform hypergraph isomorphic to $A^{(3)}(G)$. The thresholding of the weighted clique graph $C^{(3)}(G)$, retaining only edges of weight $(n-1) = 2$, produces $P^{(3)}(G)$. It is the components of this graph which are used in the clique percolation method [23] to define the communities of G.

In passing we also note that the unweighted version of the graph $A^{(n)}(G)$ is isomorphic to an n-uniform hypergraph [33, 34, 35]. Strictly $A^{(n)}(G)$ has bipartite relationships between the vertices, something not explicitly part of a hypergraph definition. However, our restriction to cliques means that the edges of the cliques can be deduced from the vertex set of each clique.

More interestingly we could project the bipartite incidence graph $B^{(n)}(G)$ onto the order n cliques to produce a new graph $C^{(n)}(G)$. I will call these "clique graphs"‡ since each vertex in these new graphs $C^{(n)}(G)$ corresponds to a clique in the original graph G. We will label the vertices of our clique graphs using the same label α we used for the cliques of G. We could define an edge in a new simple graph $L^{(n)}(G)$ between vertices α and β ($\alpha \neq \beta$) to exist if there is at least one vertex of the original graph, say $i \in \mathcal{V}$, which is common to both the order n cliques α and β. This defines an unweighted simple clique graph, $L^{(n)}(G)$, which is equivalent to the line graph the n-uniform hypergraph associated with $A^{(n)}(G)$ [34, 35]. This unweighted clique graph $L^{(n)}(G)$ captures the topology of the clique structure of G but loses a lot of other useful information.

To retain this information it makes sense to define a weighted clique graph. The simplest assignment is to set the weight of an edge between clique graph vertices α and β to be the number of vertices of G which are common to both α and β order n cliques of G. Thus our first weighted clique graph, which we will denote as $C^{(n)}(G)$, has an the adjacency matrix given by

$$C^{(n)}_{\alpha\beta} = \sum_i B^{(n)}_{i\alpha} B^{(n)}_{i\beta} (1 - \delta_{\alpha\beta}). \tag{4}$$

Note that we have also chosen to exclude self-loops, $C_{\alpha\alpha} = 0$, as for our order n clique construction the $\alpha = \beta$ case would always lead to a trivial value of n. The entries $C_{\alpha\beta}$ are therefore an integer between zero and $(n - 1)$ inclusive. The graph is undirected since $C_{\alpha\beta} = C_{\beta\alpha}$. The $C^{(3)}(G)$ clique graph for our example G is shown in figure 1.

At this point we note that the clique percolation method for finding communities [23] may be viewed as counting the connected components of an unweighted projection of this weighted clique graph $C^{(n)}(G)$ defined by using a threshold of $t = (n - 1)$ on the weights. That is an unweighted graph $P^{(n)}(G)$ with adjacency matrix

$$P^{(n)}_{\alpha\beta} = \begin{cases} 1 & \text{if } C^{(n)}_{\alpha\beta} \geq (n-1) \\ 0 & \text{if } C^{(n)}_{\alpha\beta} < (n-1) \end{cases} \tag{5}$$

So in [23] only maximal links in $C^{(n)}(G)$ are retained and the communities are then the connected communities of the resulting simple graph. This seems over restrictive since little of the information in the weights of $C^{(n)}(G)$ has been used yet many methods exist to partition weighted graphs quickly and more effectively.

However this weighted clique graph construction $C^{(n)}(G)$ appears to have a severe limitation. Each vertex $i \in \mathcal{V}$ of the original graph G contributes a total weight of $k^{(n)}_i (k^{(n)}_i - 1)/2$ to the edges of $C^{(n)}(G)$. Those which are a member of a large number of cliques (such as vertices in higher order cliques) will be giving a dominant contribution. If we want $C^{(n)}(G)$ to be a useful representation of the order n clique structure of G then it seems much better if we define a clique graph with different weights on the edges. So

‡ A better term might be "n-regular clique graphs" or "order n clique graphs" since, in the graph theory literature, the overlap between the set of *all* cliques, not just those of order n, are used to define what are also called clique graphs [4, 71]. The latter are invariably unweighted whereas weighted edges will be central in the discussion here.

we could consider the following two projections of the incidence matrix onto the cliques of G

$$D_{\alpha\beta}^{(n)} = \sum_{i,k_i^{(n)}>1} \frac{B_{i\alpha}^{(n)} B_{i\beta}^{(n)}}{k_i^{(n)} - 1} (1 - \delta_{\alpha\beta}). \tag{6}$$

$$\tilde{D}_{\alpha\beta}^{(n)} = \sum_{i,k_i^{(n)}>0} \frac{B_{i\alpha}^{(n)} B_{i\beta}^{(n)}}{k_i^{(n)}} \tag{7}$$

These adjacency matrices define weighted but undirected clique graphs, $D^{(n)}(G)$ and $\tilde{D}^{(n)}(G)$ respectively with each vertex i in the original graph G contributing $O(k_i^{(n)})$ to the weight of these graphs. These weighted line graphs have the intuitive property that the strength of a vertex α in these graphs is an integer between 1 and n, the order of the cliques being considered. For $D^{(n)}(G)$ the strength of vertex α, $s_\alpha = \sum_\beta D_{\alpha\beta}$, is the number of vertices of G which are in clique α and at least one other clique. For $\tilde{D}^{(n)}(G)$ the strength is always n, reflecting the fact that each clique has n vertices. These confirm that we are not giving any one clique too much emphasis. figure 1 shows the three weighted clique graphs, $C^{(3)}(G)$, $D^{(3)}(G)$ and $\tilde{D}^{(3)}(G)$, for our example graph.

2.2. Random Walk Motivation

There are many other definitions one might try for the weights of edges in weighted clique graphs, and as with generic bipartite graph projections, different problems may call for different definitions [36, 37, 38, 39]. However there is another way to motivate the definitions for $D^{(n)}(G)$ and $\tilde{D}^{(n)}(G)$ which suggests these are often going to be the most useful constructions.

Consider an unbiased random walk on the original graph G which takes place in two stages. First the walker moves from vertex i to any clique α for which the vertex i is a member, that is $B_{i\alpha} = 1$. All cliques attached to i are considered equally likely in an unbiased walk so this is done with probability $B_{i\alpha}^{(n)}/k_i^{(n)}$. Then the walker moves from clique α to any vertex j contained in that clique. Again all vertices in a clique are considered equally likely so this step is made with probability proportional to $B_{j\alpha}^{(n)}$. The process would be identical on the graph $A^{(n)}(G)$. It also corresponds to the natural definition of an unbiased walk on the n-regular hypergraph isomorphic to the unweighted $A^{(n)}(G)$ in which walkers move from vertex to hyperedge (the cliques here) to vertex. Finally it is the natural unbiased walk on the bipartite incidence graph $B^{(n)}(G)$. The point about the construction of $\tilde{D}^{(n)}(G)$ is that an unbiased walk on its vertices (which are the cliques of G) preserves the dynamics of the vertex-clique-vertex walk on the original graph G. Thus any analysis of the clique graph $\tilde{D}^{(n)}(G)$ using a random walk inspired measure, for example PageRank or modularity optimisation, will be equivalent to applying these measures to the order n cliques of the original graph without any bias.

By way of comparison, any vertex-vertex random walk done on $C^{(n)}(g)$ will be equivalent to a biassed vertex-clique-vertex walk on the original graph G where vertices in many order n cliques (high $k_i^{(n)}$) will be preferred by the random walker.

The one unusual point about the walk described for $\tilde{D}^{(n)}(G)$ is that it allows processes where walkers can return to the same point $i \to \alpha \to i$ and $\alpha \to i \to \alpha$. For this two-step process on an undirected graph it is in some senses natural to allow these. However, should one wish to exclude them, as is common in many cases, the definition of $D^{(n)}(G)$ corresponds to such a process.

Finally, this interpretation suggests that a factor of $1/n$ should be added to (6) and (7) to reflect the probability of moving from a clique to one of its n vertices. It is an irrelevant constant here but it will be important if one studies generalisations where cliques of different orders are considered.

3. Overlapping Communities

One can apply any of the many vertex based analysis tools to clique graphs to get non-trivial information on the cliques in the original graph. In this section we look at just one such example — the application of vertex partition methods to a clique graph.

This process assigns a unique community label to each clique in the original graph. As the first two examples are usually discussed in terms of vertices, it is natural to associate a membership function to each vertex. That is the membership of a vertex i in a community c, say f_{ic}, is given by the fraction of order n cliques containing i which are assigned to community c. That is

$$f_{ic} = \sum_\alpha \frac{B_{i\alpha}^{(n)}}{k_i^{(n)}} F_{\alpha c} \tag{8}$$

where $F_{\alpha c}$ is the membership fraction for clique α in community c. Here we have a partition of the set of n cliques so $F_{\alpha c} = \delta_{c,d}$ if clique α is assigned to community d. For simplicity, vertices which are not in any order n cliques, $k_i^{(n)} = 0$, are assigned to their own unique community. Thus vertices may be members of more than one community and the communities are generally a cover not a simple partition of the set of vertices. Note that unlike the edge partition method of [39, 40], where edges were always assigned a unique community, here edges also have a natural membership function but we will not focus on this aspect.

There are many vertex partition methods one can use. For personal convenience, the method used in the following examples is the Louvain algorithm [41] which gives values for modularity, $Q(\mathsf{A}; \gamma)$, which are close to the maximal value. We use modify the original form of modularity found in [42] and use [43]

$$Q(\mathsf{A}; \gamma) = \frac{1}{W} \sum_{C \in \mathcal{P}} \sum_{i,j \in C} \left[A_{ij} - \gamma \frac{k_i k_j}{W} \right] \tag{9}$$

where $W = \sum_{i,j} A_{ij}$ and $k_i = \sum_j A_{ij}$ is the degree of vertex i. The indices i and j run over the N vertices of the graph G whose adjacency matrix is A_{ij}. The index C runs over the communities of the partition $\mathcal{P}$. The parameter γ may be used to control the number of communities found [43].
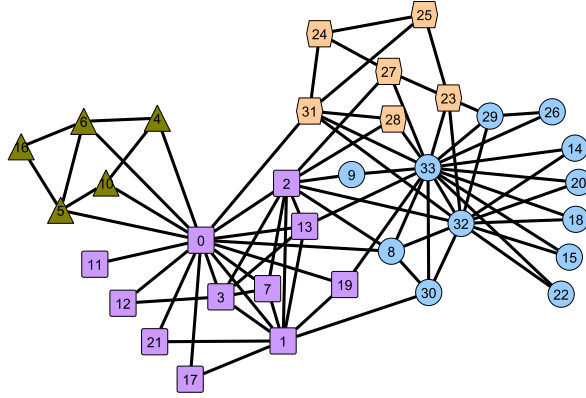
Figure 2. Zachary's Karate Club graph [46]. The colour and shape of vertices indicates the partition of the vertex set which optimises modularity Q for $\gamma = 1.0$ [48]. The number assigned to a vertex is one less than the index used by Zachary so 0 is the chief instructor, 33 the chief officer. The union of the two subsets on the left (triangles and squares), and the union of the remaining two subsets (circles and hexagons) form the two communities found by Zachary [46] using the Ford-Fulkerson binary community algorithm [47].

For instance applying the Louvain method to the original karate club graph with $\gamma = 0.3$ usually produces the same binary split found by Zachary. With $\gamma = 0.5$ the instructor's faction is split into two with the vertices $\{1, 2, 3, 7, 13\}$ assigned to their own community.

One of the big advantages of using modularity is that this may be interpreted in terms of the behaviour of random walks on the vertices [44, 45]. In this language, when we maximise modularity for the vertices of a clique graph, we can interpret this as a random walkers on the original graph moving from vertex to order to vertex and so on. However if we want unbiased walks on both the original and clique graphs, it is the $D^{(n)}$ and $\tilde{D}^{(n)}$ forms which retain a close relationship.

3.1. Karate Club

Zachary [46] gave an unweighted, undirected graph of thirty four vertices, members of a karate club. In this paper, the index of a vertex is one less than that used by Zachary [46]. Using the Ford-Fulkerson binary community algorithm [47] Zachary split this network into two factions: the instructors faction centred on vertex 0, and the officers faction, centred on the vertex numbered 33. This is shown in figure 2. Historically the club split into two distinct factions which were identical to Zachary's artificial partition except for the vertex numbered 8 here. This is identical to the actual split in the Karate club except for vertex 8.

Community algorithms which produce a partition of the vertices into two sets usually find a split similar to that of Zachary, suggesting it is an intrinsic feature of the topology of the network. Subdivisions of these sets to produce three or four communities are also often found with vertex partition methods, for example see [48, 49].

Order three cliques play a pivotal role in social network analysis (see discussion on triads in [5] and the examples of overlapping cliques in [3, 14]), so it seems logical to consider the case of $n = 3$ for the Karate Club. For $n = 2$ we would be constructing the line graphs of the Karate Club which were considered in [39]. As shown in figure 3, all but two of the thirty four vertices and all but eleven of the seventy eight edges are in order three cliques. In terms of the clique percolation protocol of [23], the order three cliques split into three clusters. Equivalently if we remove edges of weight 1 in the $C^{(3)}(G)$ graph, we are left with three components. There is one isolated order three clique, $\{24, 25, 31\}$, a second small group involving $\{0, 4, 5, 6, 10, 16\}$, and finally one massive community consisting of all the other vertices plus 0 and 31 again. These three clusters are connected in terms of all our weighted clique graphs $C^{(3)}(G)$, $D^{(3)}(G)$ and $\tilde{D}^{(3)}(G)$ but they have just one vertex in common, either 0 or 31. Thus removing the weight one edges in $C^{(3)}(g)$ is equivalent to ignoring this weak overlap, that is $P^{(3)}(G)$ has three disconnected components. Unfortunately this means that the clique percolation method of [23] fails to detect the primary binary division in this graph, one which almost all other methods successfully detect. Its only success in this context is to identify the community $\{0, 4, 5, 6, 10, 16\}$ which is often found if a community detection method can be set to find more than two communities.

The higher order cliques of the Karate Club graph are centred in the two main factions, the two percolating five cliques in $\{0, 1, 2, 3, 7, 13\}$ lie in the instructor's cluster, while the two other two non-percolating order four cliques, $(8, 30, 32, 33)$ and $(23, 29, 32, 33)$, are entirely within the officers' cluster. However the simple identification of these higher order cliques which has achieved the identification of the core of the two main factions. The percolation feature of the algorithm in [23] adds nothing. Overall, we conclude that the Karate club graph highlights the weakness of the clique percolation method [23].

However even though the order three cliques are all pervasive in this example, the basic idea of [23] and this paper that cliques can be very informative about community structure is a good one. One just needs to retain more information than is done in clique percolation and this is what the weighted clique graphs achieve.

In terms of our order three clique graphs of the karate club, applying a vertex partition algorithm to a clique graph assigns to each vertex a fractional membership of a community, f_{ic} of (8), equal to the fraction of cliques assigned to that community and which contain the given vertex.

The partition of the order three cliques into three communities found can be interpreted as overlapping communities of vertices and edges in the original Karate club graph, as shown in figure 4. For the vertices the community membership is summarised in table 3.1.

The vertices placed in the officers part of the club are placed completely in one community with the exception of vertices 2 and 8. Vertex 8 is given only 80% membership of this faction. Interestingly though most partitioning methods put this individual in the officers club, this is the one person which joined the rival faction in
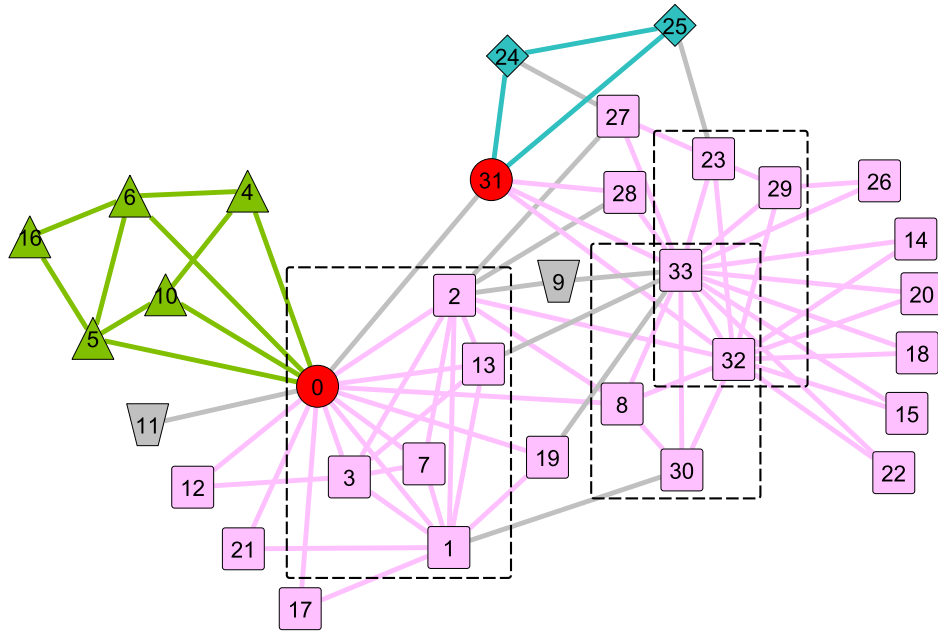
Figure 3. The Karate Club of Zachary [46]. Vertices 9 and 11 (grey trapeziums) are not in a order three clique, nor are the eleven edges $(0, 11)$, $(0, 31)$, $(1, 30)$, $(2, 9)$, $(2, 27)$, $(2, 28)$, $(9, 33)$, $(13, 33)$, $(19, 33)$, $(23, 25)$ and $(24, 27)$ (grey lines). All other vertices and edges are part of some three clique. Two vertices, 0 and 31 (shown as red circles) are the only elements in common between the three percolating order three clique clusters. These clusters are: The three clique $\{24, 25, 31\}$ (diamond shaped vertices, blue), the cluster of $\{0, 4, 5, 6, 10, 16\}$ (triangles and green), the remaining vertices and edges (square, pink) along with vertices 0 and 31. The rectangular box A contains the vertices of the two overlapping five cliques (that is $\{0, 1, 2, 3\}$ plus either 7 or 13). Boxes B and C indicate the two other non-percolating order four cliques, $\{8, 30, 32, 33\}$ and $\{23, 29, 32, 33\}$.

Community	Vertices
Instructors 1	0(78%), 1, 2(91%), 3, 7, 8(20%), 12, 13, 17, 19, 21
Instructors 2	0(22%), 4, 5, 6, 10, 16
Officers	2(9%), 8(80%), 14, 15, 18, 20, 22-33

Table 2. Overlapping Community structure of the karate club found by partitioning the vertices of $D^{(3)}(G)$ using the Louvain method with $\gamma = 0.5$. If the membership fraction, f_{ic} of table 8 is non-trivial the value is given in brackets after the index of the vertex.
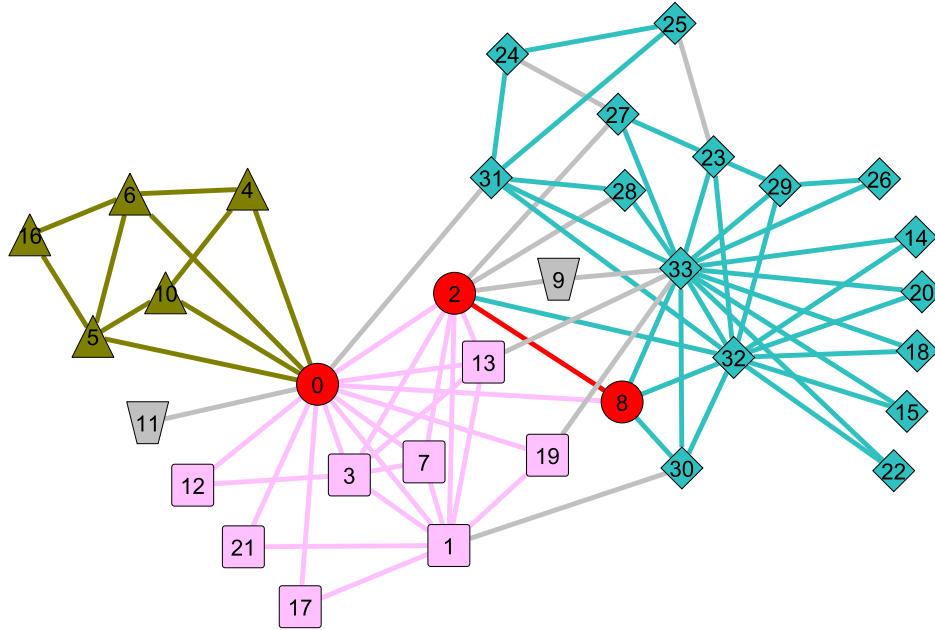
Figure 4. The Karate club shown with the partition of the order three cliques obtained by optimising modularity with $\gamma = 0.5$ on the weighted clique graph to $D^{(3)}(G)$. Three communities of order three cliques are found. Where a vertex or edge is a member of order three cliques from only one community, they are given a unique colour and vertex shape. Vertices 0, 2 and 8 (circles) and edge (2,8) are members of order three cliques in different communities and are coloured red. Finally vertices 9 and 11 (trapeziums) and edges $(0, 11)$, $(0, 31)$, $(1, 30)$, $(2, 9)$, $(2, 27)$, $(2, 28)$, $(9, 33)$, $(13, 33)$, $(19, 33)$, $(23, 25)$ and $(24, 27)$ are shown in grey as they are not part of any order three clique.

reality. Though Zachary cites special circumstances to explain this difference, he also notes that this person had only a weak affiliation to the officer's faction. It is therefore not too surprising that our method does not place this vertex in a unique community. Vertex 2 on the other hand is assigned only a 9% membership of the officers club. This individual was a strong supporter of the Instructor faction but has significant ties with members of the Officers faction. Again this does not seem an unreasonable assignment.

The instructor faction is split into two with vertices 4,5,6,10 and 16 assigned to one community while vertex 0 is given just a 22% membership of the this group. Vertex 0 has 78% of its order three cliques in the second instructor's faction which also contains all the remaining vertices with 100% membership except for vertex 2 (91%) and 8 (20%) as already discussed.

Overall the community structure found by partitioning the clique graph $D^{(3)}(G)$ reflects the true nature of the karate club extremely well.

In the same way we can also study the vertex partitioning of the clique graph $C^{(3)}(G)$ for the karate club. This weighted clique graph we expected to give too much emphasis to vertices which are members of many cliques, typically the high degree vertices. However we find that applying the Louvain method with $\gamma = 0.5$ to partition the vertices of $C^{(3)}(G)$ that we end up with two communities. In terms

of the original vertices of the karate club, these are exactly the same as found with $D^{(3)}(G)$ but where the two instructors communities have been merged. Thus though this is still an overlapping community structure, the overlap (vertices 2 and 8 again) is weak as indicated in table 3.1. So the community structure derived from $C^{(3)}(G)$ is also consistent with the binary split of Zachary.

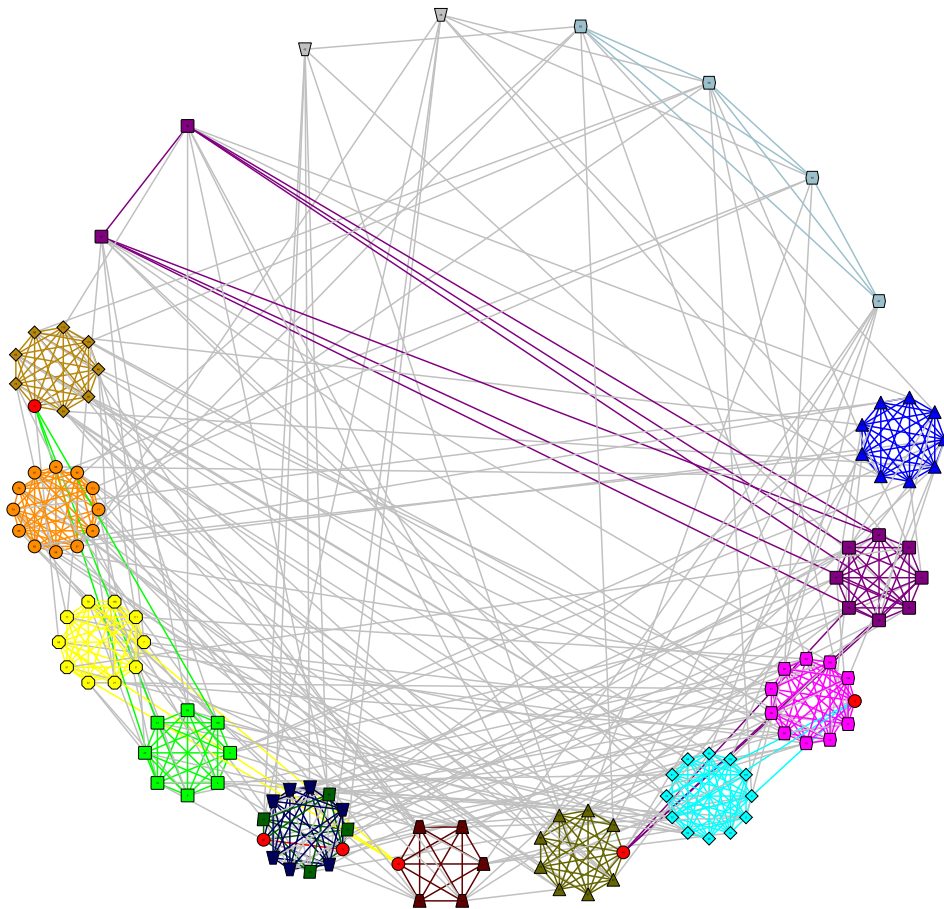| Community	Vertices
Instructors	0, 1, 2(91%), 3-7, 8(20%), 10, 12, 13, 16, 17, 19, 21
Officers	2(9%), 8(80%), 14, 15, 18, 20, 22-33

Table 3. Overlapping Community structure of the karate club found by partitioning the vertices of $C^{(3)}(G)$ using the Louvain method with $\gamma = 0.5$. f the membership fraction, f_{ic} is non-trivial the value is given in brackets after the index of the vertex. The binary partition found by Zachary [46] using the Ford-Fulkeson method [47] is identical if we assign vertices 2 and 8 completely to the community with which they have the largest overlap, the Instructors and the Officers respectively.

3.2. American College Football Network

Another example that has been used elsewhere [42] is the network formed by teams in a league with each vertex representing one team with two teams linked if they have played each other that season. For instance of the 115 teams in the American College Football Division 1-A in the 2000 season, all but eight are organised into eleven conferences of various sizes§. As teams played between 7 and 13 games with an average of 10.7 games, most teams do not play each other. However if a team is in a conference then they play the majority of their games against other teams in the same conference. For this reason the eleven conferences are readily apparent as eleven tightly knit subgraphs, each of which contains cliques of order five or higher making it a useful test for community detection methods.

The results using order four cliques are very good. Using percolation, shown in figure 5, or vertex partitioning of either the $C^{(4)}$ or $D^{(4)}$ clique graphs (optimising modularity with $\gamma = 1$) gives almost the same results. This is that each conference corresponds to one, or in one case, two communities. There is a little overlap, almost all teams are part of four cliques which involve only teams in their conference. The exception is that two independents are deemed part of the community centred on one of the conferences. A final community is an isolated clique of four independents. The only difference between the approaches is that one conference (the seventh counting clockwise from the conference at 3 o'clock) is split into its two divisions with percolation and $D^{(4)}$ but not with $C^{(4)}$.

§ The conference assignment used in [42] appears to be for the 2001 season. The data used here for the games played between two teams is based on the file `football.gml` downloaded from Newman's website which is associated with [42]. However the conference assignments used here have been derived from other sources. See Appendix C.2 for additional information.
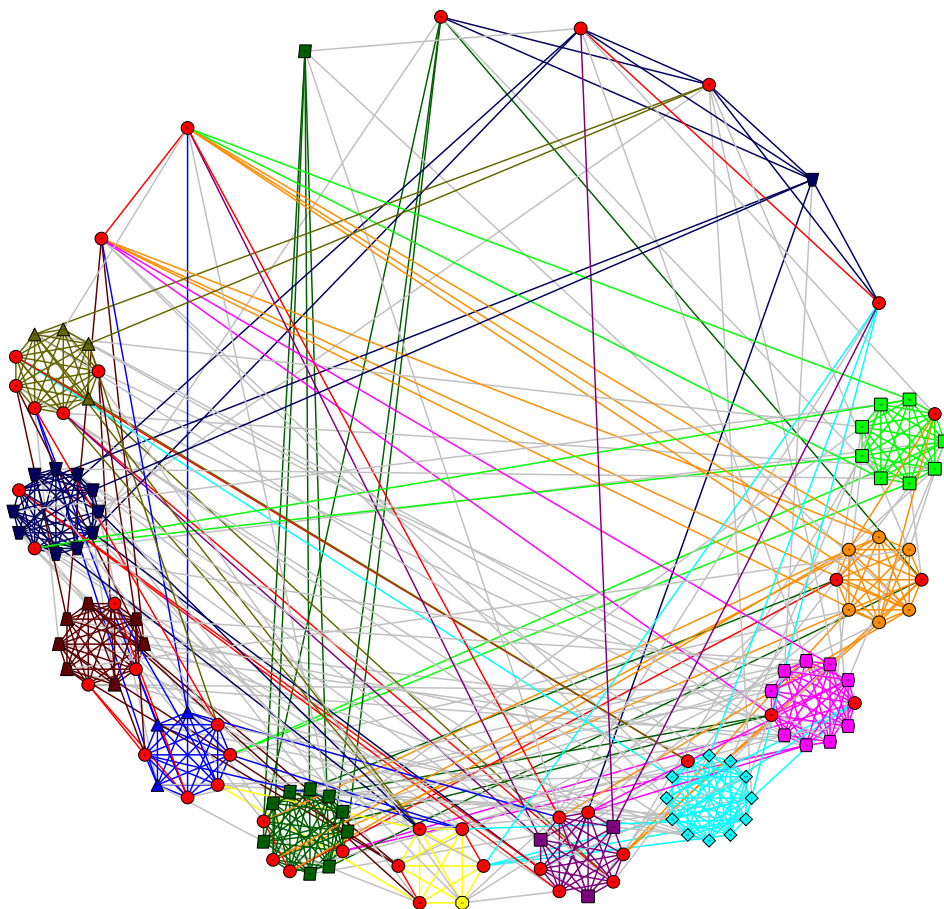
Figure 5. Network based on games played in 2000 between teams of the Division 1-A American College Football league. The community structure is found using order four clique percolation. Edges and vertices in a unique community are shown in a colour unique to that community. Vertices in the same community are also shown using the same shape (some shapes are used for two distinct communities). Vertices and links not in a order four clique are shown in grey. Vertices and edges in more than one community are shown in red and using circle for the vertices. The teams of each conference are placed in a small circle which are in turn located around a large circle (see table (C1)). The eight independents appear as single vertices around the large circle. The community structure detected by order four clique percolation matches the conference structure almost perfectly. Note the conference at about 7 o'clock is split into its two divisions.

Looking at 5-cliques vertex partitioning of the $C^{(5)}$ and $D^{(5)}$ clique graphs with $\gamma = 1.0$ gives the same structure, putting all but one team into the correct conference though now two conferences are split into their divisions. Percolation does almost as well but one of the conferences is split into three parts. However as n is raised further, results get rapidly worse and whole conferences fail to be identified. This is simply because these higher order cliques are much rarer in this data set.

The real test comes when we consider three cliques for this American College Football network. This is a disaster for the percolation approach as only four communities are identified, two correspond to one conference each, one is based on

Figure 6. Network based on games played in 2000 between teams of the Division 1-A American College Football league. The vertices are placed in the same locations as figure 5. The community structure detected by vertex partitioning of the order three clique graph $C^{(3)}(G)$ clearly identifies the teams in each conference. About one-third of conference teams are members of communities containing teams from other conferences. However the majority of triangles containing conference teams contain only other teams in the same conference and so conference identification is simple.

the clique of four independents and all the remaining conference teams are all in one giant community. However vertex partitioning of both the $C^{(3)}$ and $D^{(3)}$ clique graphs still works, see figure 6. About two-thirds of the conference teams are placed in a unique community containing only teams from their conference and perhaps some independents. For the other third, it is still true that the vast majority of triangles (at least 79%) containing these conference teams contain only other teams in the same conference. That is it is easy to classify the overlap as weak and accurate conference identification remains simple. Even the associations seen between some of the independents remain clear at the 75% level.
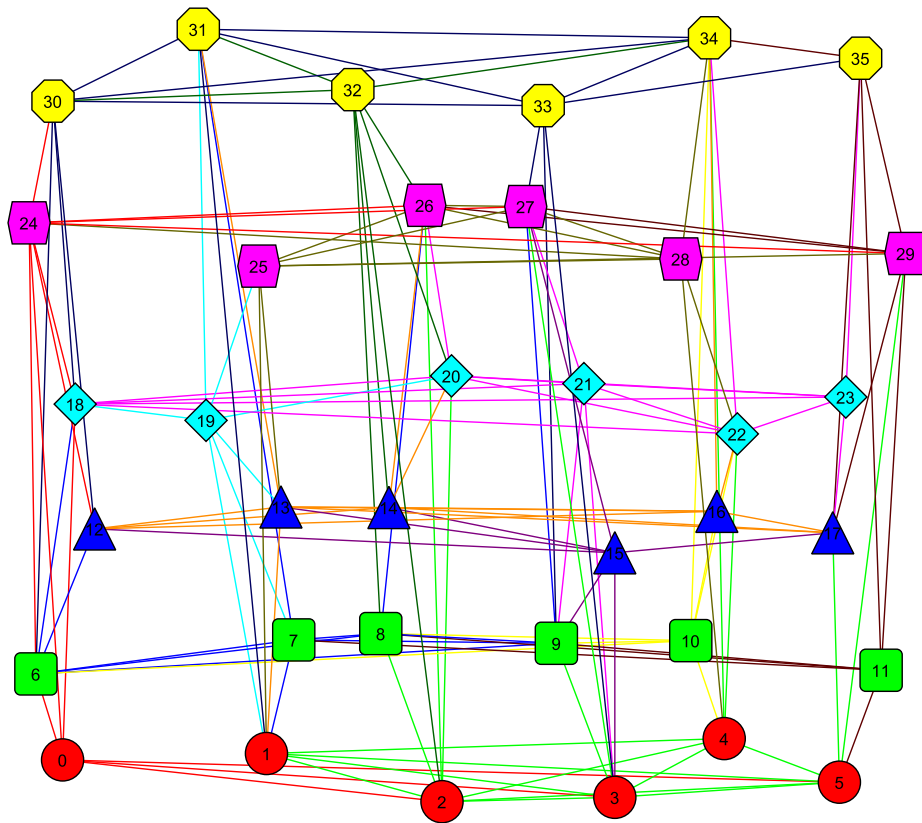
3.3. Benchmark Graph

The previous examples have shown that community detection based on the vertex partitioning of a clique graph can be very successful, much more than simple clique percolation. However all the examples above are well known from the literature which is dominated by successful methods for finding vertex partitions of graphs. This means that exemplary networks drawn from the literature are likely to have an inherent bias towards those that give 'good' results for most vertex partitioning schemes such as the Louvain method used here.

In fact the situation with these standard examples may be even more complicated. The definition of a 'good' community is usually taken to be in terms of some reference vertex partition, Zachary's original split [46] into two vertex sets using the Ford-Fulkerson binary community algorithm [47], or the association of American College football teams with their conferences. A 'good' method is defined to be one which obtains results close to these externally specified partitions. Indeed this is what has been done to judge the clique graph method a success on the previous examples. However one might argue that a good overlapping community structure might reveal subtleties missed by simple vertex partitions. For instance, it is clear that the Instructor in the Karate club example (vertex 0) is a member of two distinct communities, and indeed is the only connection between the two. In this sense the reference partition of the vertices may well not be the 'best' way to describe the community structure in a network. Unfortunately, it is often not possible to produce a 'better' reference community structure. Either the data to do this is not available or there is still a subjective element to any definition of a better community structure.

Therefore the final example is an artificial benchmark graph constructed to reflect the overlapping community structures expected in many situations. Thirty six vertices are placed on a square grid. Each vertex is visited in turn and two more vertices are chosen at random, subject to the constraints that the vertices are distinct, and that all three vertices are either all in the same row or they are all in the same column. The three vertices are then connected forming a triangle, using any existing edge or adding more if needed. Once all thirty six vertices have been visited we repeat until the desired number of triangles have been added. This produces a simple graph. where every edge and every vertex is part of at least one triangle.

This benchmark graph can be thought of as a group of thirty six individuals who work in six different firms and are members of one of six different social groups (e.g. common sports team, extended family group) outside work. No two individuals both work at the same firm and have the same social interests. Of course this last restraint is somewhat artificial and the square grid is too simplistic, imposed for purely for visualisation purposes. Nevertheless it does try to capture the idea that people are members of more than one community and their social interactions, here represented by the triangles, may take place in different communities. These communities may not be obvious if one studies just the existence of bilinear relationships (e.g. edges only indicate

Figure 7. A vertex partition and an edge partition of the benchmark graph in which 72 triangles are placed. The edge partition is found by finding a vertex partition of $D^{(2)}(G)$ — the weighted line graph $D(G)$ described in [39]. Both partitions are found using the Louvain algorithm to maximise modularity (9) with $\gamma = 1.2$. On this run, the vertex partition finds the six communities associated with the rows, indicated by the vertex shapes and colours, but the row communities are completely missed. The edge partition finds eleven communities. Most of the lines in each column and most of this n two of the rows are correctly assigned to a single community, as indicated by the edge colours.

phone calls were made or emails were sent) rather than analysing the nature of each contact. The aim for a community detection method is to find the twelve communities, one for every row and one for every column.

Any vertex partition method will fail to find at least half the structure. In the example we have used (the Louvain algorithm applied directly to the vertices) it does seem to be relatively successful, usually finding 6 or 7 communities, a good approximation to either the rows or the columns, for $\gamma = 0.6$ to 2.0 in (9). One good example is shown in figure 7.

Perhaps surprisingly partitioning the edges by making a partition of the vertices in the graph $D^{(2)}(G)$ (the weighted line graph $D(G)$ in [39]) is not much more successful. In principle this should also be able to detect the overlapping communities. The problem here may be that there are also many rectangles in this artificial benchmark and these
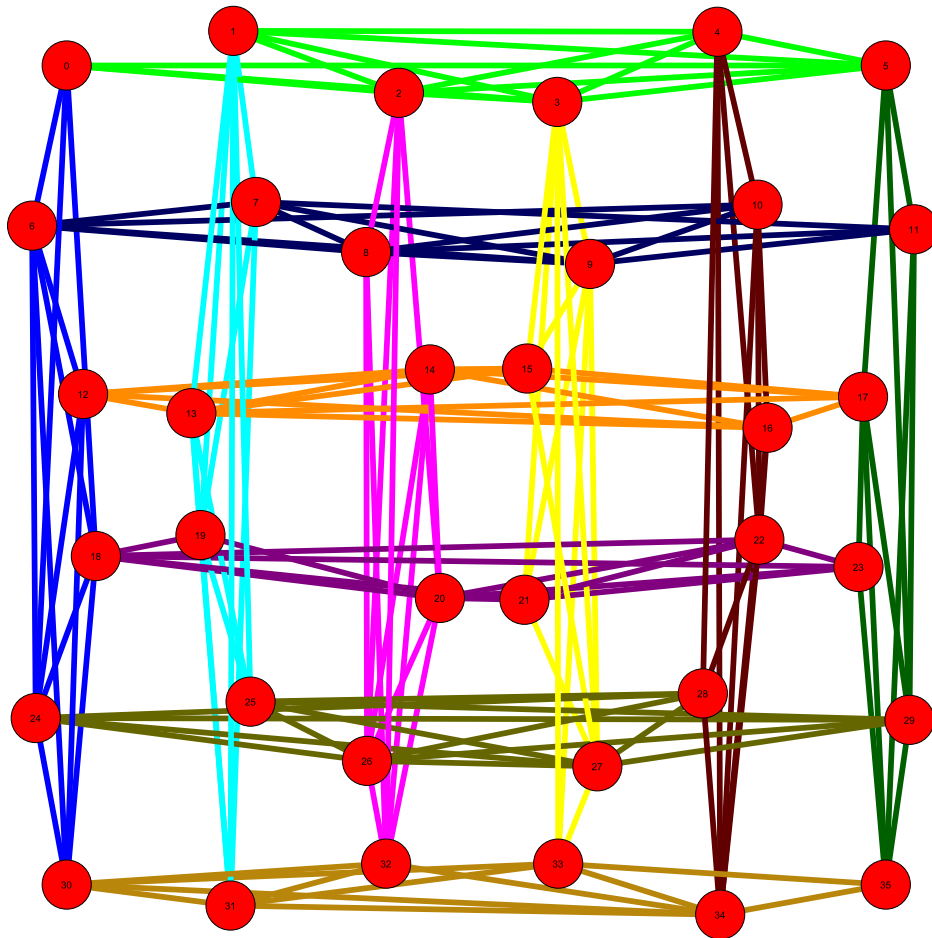
Figure 8. The community structure based on the partition of order three cliques of the same benchmark graph as in figure 7. Produced by applying the Louvain algorithm to the $C^{(3)}$ clique graph, maximise modularity with $\gamma = 3.0$. Twelve communities associated with the columns are found matching the column and row communities perfectly. This is indicated by the edges in each row having a unique colour, and similar for the columns.

be important when optimising modularity in the weighted line graph.

On the other hand the clique detection method is almost perfect. Looking at the three cliques, and applying the Louvain method to both $C^{(3)}$ and $D^{(3)}$ clique graphs, both the column and vertex structure is found almost perfectly as shown in figure 8. The clique percolation method is also perfect on this benchmark graph.

4. Discussion

The aim of this paper has been to show that if one wishes to focus on the role of cliques in a graph G, one may encode this information as a graph, a clique graph whose vertices represent the cliques in the original graph G. The advantage is that there are many well established methods for analysing the properties of vertices of a graph and for the price of a simple transformation, these can be applied to obtain the same information about

the cliques. It avoids the natural bias towards vertices found in network analysis while exploiting the same bias by working with clique graphs in order to move the focus onto the cliques of the original graph G.

One of the most important differences between this work and previous research is that the emphasis here is on the cliques. Other studies of clique overlap usually retain the focus on the original vertices and use constructions similar to the $A_{ij}^{(n)}$ of table 3. That is the vertices are still the same as the original graph but now the edge weights carry the information about clique overlap. The emphasis here and in [23] (and indeed in the clique graphs of [4, 71]) is to exploit our vertex centric view of graphs and to use a new graph where the vertices represent the cliques of the original graph.

The construction of a clique graph is not unique, several definitions of weighted clique graph are suggested here, motivated by work on useful projections of bipartite graphs (for example see [36, 37, 38]) and on the case of order two cliques, the line graphs of [39, 40]. As emphasised in [39, 40] the construction of $D^{(n)}$ (6) has the advantage that a random walk on its vertices retains the dynamical structure of random walks on the vertices of either the bipartite graph B or the original graph G.

The most obvious limitation so far is that our original graph G must be simple. However it is straightforward to define a second weighted bipartite graph where the entry in the adjacency matrix $\bar{B}_{i\alpha}$ is the weight of the clique α. There are many ways to define the weight of a clique based on the weights of the edge, for example see [21, 50, 51]. We would consider replacing our definition of the adjacency matrix of the weighted clique graph $D^{(n)}$ of (6) by

$$D_{\alpha\beta}^{(n)} = \sum_{i,k_i^{(n)}>1} \frac{\bar{B}_{i\alpha}^{(n)} B_{i\beta}^{(n)}}{s_i^{(n)} - \bar{B}_{i\alpha}^{(n)}}(1 - \delta_{\alpha\beta}). \tag{10}$$

Here $s_i^{(n)} = \sum_i \bar{B}_{i\alpha}$ and $B_{i\alpha}$ is equal to one (is zero) only if $\bar{B}_{i\alpha}$ is non-zero (is zero). This form is again motivated by considering a random walk that moves from vertex i to clique α to vertex etc. This approach was used for line graphs ($n = 2$) in [40].

An important difference between this work and much of the literature is that I have focused on all cliques of a fixed order n. This can reflect the importance of one particular clique in a given context. For instance the triad plays an important role in social network analysis [3, 5, 9, 10, 11, 12, 14, 16]. In other circumstances choosing the order of cliques used may just be a useful computational freedom, as here and in [23]. However it is straightforward to generalise all the constructions to a situation where the cliques are drawn from a different set of cliques $\mathcal{C}$, containing cliques of different orders. We just define a new bipartite incidence matrix $B_{i\alpha}$ which is one (zero) if vertex $i \in \mathcal{V}$ is in clique $\alpha \in \mathcal{C}$ which is now drawn from some more general set of cliques $\mathcal{C}$. The clique overlap graph $A^{(n)}(G)$ defined in (3) is replaced by

$$A_{ij}(G, \mathcal{C}) = \sum_{\alpha \in \mathcal{C}} B_{i\alpha} B_{j\alpha}(1 - \delta_{ij}), \qquad \forall\, i, j \in \mathcal{V}. \tag{11}$$

In fact most work on the overlap of cliques in a graph is based on $A(G, \mathcal{C}^{(\mathrm{max},n)})$ where $\mathcal{C}^{(\mathrm{max},n)}$ is the set of all maximal cliques whose order is at least n, for instance see

[4, 6, 7, 8, 13, 15]. In principle we could generalise $C^{(n)}(G)$ $D^{(n)}(G)$ and $\tilde{D}^{(n)}(G)$ of equations (4) (6) and (7) in the same way, e.g. the 'co-clique' graph defined in [8] would be $C(G, \mathcal{C}^{(\max,n)})$. However the random walk argument suggests that the definition of $D^{(n)}(G)$ should now be

$$D_{\alpha\beta}(G,\mathcal{C}) = \sum_{i,k_i^{(n)}>1} \frac{B_{i\alpha}}{(k_i^{(n)}-1)} \frac{B_{i\beta}}{n_\beta} (1-\delta_{\alpha\beta}), \qquad \alpha,\beta \in \mathcal{C} \qquad (12)$$

where $n_\beta = \sum_i B_{i\beta}$ is the order of the clique β.

It has been argued that considering only complete subgraphs is too 'stingy' [52]. So we may be interested in the case where α is a subgraph of G isomorphic to one of a small set of more general motifs, subgraphs which are not necessarily regular graphs. Interesting examples would be those representing cohesion, such as the n-cliques, n-clans, n-clubs, k-plex and k-core structures used in social networks and elsewhere [5, 11, 14]. The incidence matrix $B_{i\alpha}$ may be defined as before but for this new set of subgraphs and it can be projected onto vertices or motifs to capture motif overlap. For instance the generalisation of A and C graphs from a set of maximal cliques of minimum order as used in [6], to equivalents for a set of motifs was given in [32]. By the time we have reached this level of complexity we are essentially defining hypergraph structures on the set of vertices. On the other hand, such motif graph constructions are still useful ways to convey the motif overlap information and, by using a graph to do so, standard tools may be used to analyse this information.

Such generalisations also suggest how these clique graph constructions could be adapted for a directed or signed graphs. In these cases there are many different ways of having connections between, say, three vertices but we can just keep the relevant motifs, e.g. using the set of triangles regarded as being balanced in balance theory [5, 72].

In all this work we have always considered the overlap of vertices and motifs. If the fundamental structure is a graph G then, in the spirit of [39, 40], we may want to define overlap in terms of the edges of G. Thus $B_{e\alpha}$ is one if edge e is part of motif α. As an example consider a regular square lattice as the original graph G and suppose we take a unit square as the motif of interest. It is simple to see that the motif graphs formed using the edge overlap, $\sum_e B_{e\alpha}B_{e\beta}(1-\delta_{\alpha\beta})$ etc, are also square lattices, i.e. in term of topology these edge-motif graphs are just the dual lattice.

Finally we have illustrated one use for clique graphs, that of detection of overlapping communities, a cover and not a partition of the original vertices. There has been a recent surge in interest in this problem, for instance see [23, 53, 54, 55, 56, 57, 58, 59, 39, 25, 60, 61, 35, 69, 62, 63, 64, 65, 66, 40, 67]. By way of contrast, the literature on social network analysis, where clique overlap is better known, is almost entirely focused on cohesive subgroups which are partitions of the original vertices, for instance in [46, 3, 6, 10, 32, 17, 13]. This follows in part because of the focus in this area on the graphs which retain the original vertices such as the A's of (3) and (11). Since most algorithms produce a partition of the vertices, such as the Johnson Hierarchical Clustering Scheme [70] (as used in UCInet [13]), non-overlapping vertex communities

are the norm in this area.

The approach suggested here has the advantage that for the price of a simple transformation to produce a clique graph $C(G,\mathcal{C})$ or $D(G,\mathcal{C})$, the much more extensive work on vertex partition of a graph may be applied to produce an overlapping community structure without additional work. This may reduce the development time for a project. In terms of computational efficiency, the clique graphs are generally bigger but by how much depends on the detailed structure of the graph. The speed savings of a good fast vertex partitioning algorithm, such as [41, 68], may compensate for the larger size of the clique graph. The most important feature though is that this method puts the emphasis on cliques. It is likely that this approach will be better than other methods when cliques play a key role. For instance, it was noticeable that in the benchmark network created out of triangles, edge partitioning, an alternative overlapping community technique, was not nearly as effective as the vertex partitioning of clique graphs.

Bibliography

[1] Evans T 2004 *Contemporary Physics* **45** 455
[2] Luce R and Perry A 1949 *Psychometrika* **14** 95
[3] Freeman L C 1992 *The Sociological Concept of "Group": An Empirical Test of Two Models*, American Journal of Sociology **98** 152
[4] Harary F 1994 *Graph Theory* (Addison-Wesley)
[5] Wasserman S and Faust K 1994 *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)* (Cambridge University Press)
[6] Freeman L C 1996 *Cliques, Galois lattices, and the structure of human social groups*, Social Networks **18** 173
[7] Provan K G and Sebastian J G 1998 *Networks within Networks: Service Link Overlap, Organizational Cliques, and Network Effectiveness* The Academy of Management Journal **41** 453
[8] Everett M G and Borgatti S P 1998 *Analyzing Clique Overlap* Connections **21** 49
[9] Krackhardt D 1998 Simmelian Ties: Super Strong and Sticky *Power and Influence in Organizations* (Thousand Oaks: CA: Sage) pp 21
[10] , Krackhardt D 1999 *The Ties That Torture: Simmelian Tie Analysis in Organizations*, Research in the Sociology of Organizations **16** 183
[11] Scott J 2000 *Social Network Analysis: A Handbook* 2nd ed (London: Sage Publications)
[12] Krackhardt D and Kilduff M 2002 *Structure, culture and Simmelian ties in entrepreneurial firms*, Social Networks **24** 279
[13] Borgatti S P, Everett M G and Freeman L C 2002 *UCInet02 for Windows: Software for Social Network Analysis* (Harvard, MA: Analytic Technologies URL `http://www.analytictech.com/UCInet02/`)
[14] de Nooy W, Mrvar A and Batagelj V 2005 *Exploratory Social Network Analysis with Pajek* Structural Analysis in the Social Sciences (No. 27) (Cambridge University Press)
[15] Hanneman R A and Riddle M 2005 *Introduction to social network methods*, (University of California, Riverside, published in digital form at URL `http://faculty.ucr.edu/$\sim$hanneman/`)
[16] Krackhardt D and Handcock M 2007 Heider vs Simmel: Emergent Features in Dynamic Structure *The Network Workshop Proceedings. Statistical Network Analysis: Models, Issues and New Directions*, (New York: Springer) pp 14
[17] Bellotti E 2009 *Methodological Innovations Online* **4** 53
[18] Granovetter M S 1973 *The Strength of Weak Ties*, American Journal of Sociology textbf78 1360

[19] Granovetter M S 1983 *The Strength Of Weak Ties: A Network Theory Revisited*, Sociological Theory **1** 201

[20] Bron C and Kerbosch J 1973 *Commun. ACM* **16** 575–577

[21] Samudrala R and Moult J 1998 *J. Molecular Biology* **279** 287

[22] Takemoto K, Oosawa C and Akutsu T 2007 *Physica A* **380** 665

[23] Palla G, Derenyi I, Farkas I and Vicsek T 2005 *Nature* **435** 814

[24] Kumpula J M, Kivelä M, Kaski K and Saramäki J 2008 *Phys. Rev. E* **78** 026109

[25] Yan B and Gregory S 2009 *Detecting communities in networks by merging cliques*, *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS 2009)* (IEEE) pp 832

[26] Takemoto K and Oosawa C 2005 *Phys. Rev. E* **72** 046116

[27] Toivonen R, Onnela J P, Saramaki J, Hyvonen J and Kaski K 2006 *Physica A* **371** 851

[28] Fortunato S 2010 *Physics Reports* **486** 75

[29] Porter M A, Onnela J P and Mucha P 2009 *Notices of the American Mathematical Society* **56** 1082

[30] Kossinets G and Watts D 2006 *Empirical Analysis of an Evolving Social Network*, *Science* **311** 88

[31] Kumpula J M, Onnela J P, Saramaki J, Kaski K and Kertesz J 2007 *Phys. Rev. Lett.* **99** 228701

[32] Falzon L 2000 *Determining groups from the clique structure in large social networks*, Social Networks **22** 159

[33] Cattuto C, Schmitz C, Baldassarri A, Servedio V D P, Loreto V, Hotho A, Grahl M and Stumme G 2007 *AI Commun.* **20** 245

[34] Neubauer N and Obermayer K 2009 Hyperincident connected components of tagging networks *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia* (New York, NY, USA: ACM) pp 229

[35] Neubauer N and Obermayer K 2009 Towards community detecion in k-partite, k-uniform hypergraphs

[36] Newman M E J 2001 *Phys. Rev. E* **64** 016131

[37] Guillaume J L and Latapy M 2006 *Physica A* **371** 795

[38] Zhou T, Ren J, Medo M and Zhang Y C 2007 *Phys. Rev. E* **76** 046115

[39] Evans T and Lambiotte R 2009 *Phys. Rev. E* **80** 016105

[40] Evans T S and Lambiotte R 2010 Line graphs of weighted networks for overlapping communities (Preprint `arXiv:0912.4389`)

[41] Blondel V D, Guillaume J L, Lambiotte R and Lefebvre E 2008 *J. Stat. Mech.* P10008

[42] Girvan M and Newman M E J 2002 *PNAS* **99** 7821

[43] Reichardt J and Bornholdt S 2006 *Phys. Rev. E* **74** 016110

[44] Delvenne J C, Yaliraki S N and Barahona M 2008 Stability of graph communities across time scales (Preprint `arXiv:0812.1811`)

[45] Lambiotte R, Delvenne J C and Barahona M 2008 (Preprint `arXiv:0812.1770`)

[46] Zachary W 1977 *Journal Of Anthropological Research* **33** 452—473

[47] Ford L R and Fulkerson D R 1956 *Canadian Journal of Mathematics* **8** 399

[48] Agarwal G and Kempe D 2007 (Preprint `arXiv:0710.2533`)

[49] Cheng X Q and Shen H W 2010 *J. Stat. Mech.* P04024

[50] Palla G, Farkas I J, Pollner P, Derenyi I and Vicsek T 2007 *New Journal of Physics* **9** 186

[51] Farkas I, Abel D, Palla G and Vicsek T 2007 *New Journal of Physics* **9** 180

[52] Alba R 1973 *Journal of Mathematical Sociology* **3** 113

[53] Baumes J, Goldberg M and Magdon-Ismail M 2005 Efficient identification of overlapping communities *In IEEE International Conference on Intelligence and Security Informatics (ISI* pp 27

[54] Li X, Liu B and Yu P 2006 Discovering overlapping communities of named entities *PKDD 2006, LNAI 4213* (*Lecture Notes in Computer Science* vol 4213) ed Fürnkranz J, Scheffer T and Spiliopoulou M (Springer-Verlag Berlin Heidelberg) pp 593

[55] Zhang S, Wang R S and Zhang X S 2007 *Physica A* **374** 483

[56] Nicosia V, Mangioni G, Carchiolo V and Malgeri M 2009 *J. Stat. Mech.* P03024

[57] Gregory S 2008 An algorithm to find overlapping community structure in networks *Machine Learning and Knowledge Discovery in Databases: Proceedings of 18th European Conference on Machine Learning (ECML) and the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)* (Springer) pp 91

[58] Gregory S 2008 A fast algorithm to find overlapping communities in networks *Machine Learning and Knowledge Discovery in Databases: Proceedings of 18th European Conference on Machine Learning (ECML) and the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)* (Springer) pp 408

[59] Ahn Y Y, Bagrow J P and Lehmann S 2010 *Communities and hierarchical organization of links in complex networks*, Nature 1038 1 [`arXiv:0903.3178`]

[60] Lázár A, Ábel D and Vicsek T 2009 Modularity measure of networks with overlapping communities (Preprint `arXiv:0910.5072`)

[61] Lancichinetti A, Fortunato S and Kertesz J 2009 *New Journal of Physics* **11** 033015

[62] Pizzuti C 2009 Overlapped community detection in complex networks *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation* (New York, NY, USA: ACM) pp 859

[63] Sawardecker E N, Sales-Pardo M and Amaral L A 2009 *E. Phys. J. B* **67** 277

[64] Shen H W, qi Cheng X and Guo J F 2009 *J. Stat. Mech.* P07042

[65] Wang X F and Liu Y B 2009 *Journal of University of Electronic Science and Technology of China* **38** 537.

[66] Wei F, Qian W, Wang C and Zhou A 2009 *World Wide Web* **12** 235

[67] Shang M S, Chen D B and Zhou T 2010 *Chinese Physics Letters* **27** 058901

[68] Rosvall M and Bergstrom C T 2008 *PNAS* **105** 1118

[69] Evans T S and Plato A D K 2007 Phys.Rev.E **75** 056101

[70] Johnson S 1967 *Hierarchical clustering schemes*, Psychometrika **32** 241

[71] Weisstein E W Clique Graph *MathWorld–A Wolfram Web Resource* URL `http://mathworld.wolfram.com/CliqueGraph.html`

[72] Szell M, Lambiotte R and Thurner S, *Multirelational organization of large-scale social networks in an online world*, PNAS

Appendix A. Alternative Frequency Count

Word	Rank	Count	Word	Rank	Count
network	8	161	simple	68	33
vertices	15	104	distribution	82	27
networks	17	90	scale	85	26
random	20	86	connected	91	22
degree	24	71	edge	106	20
graph	25	71	links	108	20
edges	28	66	neighbours	109	20
lattice	29	64	hubs	121	17
power	30	64	scale-free	124	17
vertex	34	61	clustering	129	16
distance	55	39	regular	155	14
small	64	35	graphs	207	10
world	66	34	link	252	8

Table A1. Table showing the frequencies of all the network related words in the review of networks [1]. In calculating the frequencies, all words from the text were included, with no alterations. The rank is by the number of occurrences of each word, with alphabetical order used to order words with equal counts.

Appendix B. Karate Club

Appendix B.1. Cliques of the Karate Club

The index of each vertex is one less than that used by Zachary [46], so run from 0 to 33.

- The highest order of a clique which contains either vertex 9 or 11 is two.
- All other vertices are in order three cliques. One three clique $\{24, 25, 31\}$ has no edges in common with other order three cliques and only vertex 31 is in other order three cliques. The remaining 3 cliques split into two groups. Vertices $\{0, 4, 5, 6, 10, 16\}$ form a percolating cluster of order three cliques. The remaining vertices (all those not mentioned in the list so far) along with vertices 0 and 31 form a second percolating cluster. Note this means that eleven edges $(0, 11)$, $(0, 31)$, $(1, 30)$, $(2, 9)$, $(2, 27)$, $(2, 28)$, $(9, 33)$, $(13, 33)$, $(19, 33)$, $(23, 25)$ and $(24, 27)$ are also not part of any order three clique but every other edge is part of a triangle.
- There are two order four cliques that are not sub graphs of cliques of order 5. These are $(8, 30, 32, 33)$ and $(23, 29, 32, 33)$. Note these are not percolating as they share two rather than three vertices in common.
- There are two order five cliques which are percolating around the common four clique of $\{0, 1, 2, 3\}$. Vertices 7 and 13 make up the two five cliques in this case.
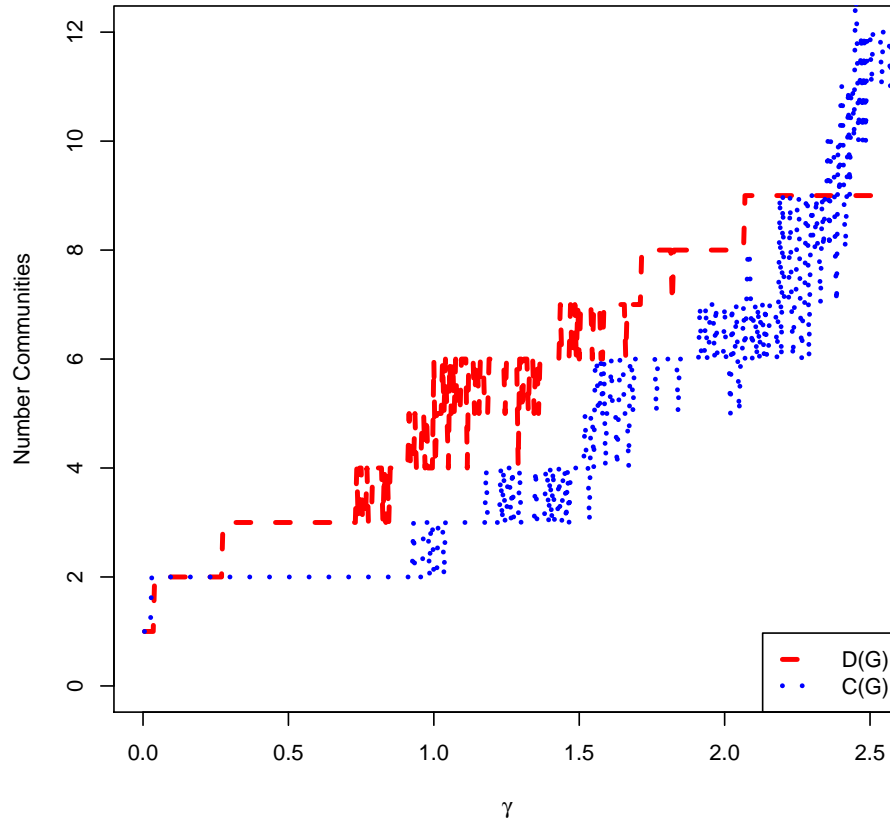
Figure B1. The number of communities found maximising the modularity of (9) of clique graphs of Zachary's Karate Club graph as γ is varied. For the clique graphs $C^{(3)}$ (blue dotted line) and $D^{(3)}$ (red dashed line).

Appendix B.2. Communities of the Karate Club

As we see from figure B1 the stability in the number of communities found by the Louvain method suggests that $\gamma = 0.5$ is a reasonable value to study. In this case we find three communities of cliques in $D^{(3)}(G)$. These correspond closely to the three communities found for a Louvain partition of the vertices with $\gamma = 0.3$, which in turn is fully consistent with the binary split of Zachary.

The communities referred to in the text are summarised in the following tables B1, B2, B3 and B4.

Community	Vertices
Instructors	0-7, 10-13, 16, 17, 19, 21
Officers	8, 9, 14, 15, 18, 20, 22-33

Table B1. The binary partition found by Zachary [46] using the Ford-Fulkeson method [47]. Note that vertex 8 that is the only one placed in the wrong faction as compared to the actual split in the club.

Community	Vertices
Instructors 1	0-3, 7, 11-13, 17, 19, 21
Instructors 2	4, 5, 6, 10, 16
Officers 1	8, 9, 14, 15, 18, 20, 26, 29, 30, 32, 33
Officers 2	23, 24, 25, 27, 28, 31

Table B2. The vertex partition of the Zachary karate club [46] which produces the largest modularity $Q(A)$ [48].

Community	Vertices
Instructors	0, 1, 2(91%), 3-7, 8(20%), 10, 12, 13, 16, 17, 19, 21
Officers	2(9%), 8(80%), 14, 15, 18, 20, 22-33

Table B3. Overlapping community structure of the karate club found by partitioning the vertices of $C^{(3)}(G)$ using the Louvain method with $\gamma = 0.5$. If the membership fraction, f_{ic} of (8), is non-trivial the value is given in brackets after the index of the vertex.

Community	Vertices
Instructors 1	0(78%), 1, 2(91%), 3, 7, 8(20%), 12, 13, 17, 19, 21
Instructors 2	0(22%), 4, 5, 6, 10, 16
Officers	2(9%), 8(80%), 14, 15, 18, 20, 22-33

Table B4. Overlapping community structure of the karate club found by partitioning the vertices of $D^{(3)}(G)$ using the Louvain method with $\gamma = 0.5$. If the membership fraction, f_{ic} of (8), is non-trivial the value is given in brackets after the index of the vertex.

Another example of the community structure found on the karate club graph, figure B2.
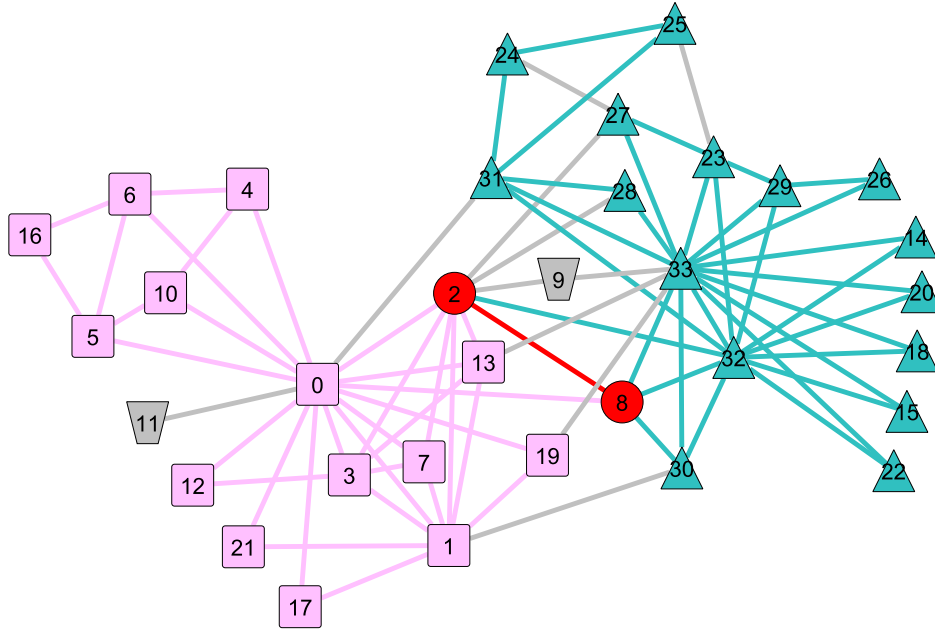
Figure B2. The Karate club shown with the partition of the order three cliques obtained by optimising modularity with $\gamma = 0.5$ on the weighted clique graph to $C^{(3)}(G)$. Two communities of order three cliques are found. Where a vertex or edge is a member of order three cliques from only one community, they are given a unique colour and vertex shape. Vertices 0, 2 and 8 (circles) and edge (2,8) are members of order three cliques in different communities and are coloured red. Finally vertices 9 and 11 (trapeziums) and edges $(0, 11)$, $(0, 31)$, $(1, 30)$, $(2, 9)$, $(2, 27)$, $(2, 28)$, $(9, 33)$, $(13, 33)$, $(19, 33)$, $(23, 25)$ and $(24, 27)$ are shown in grey as they are not part of any order three clique.
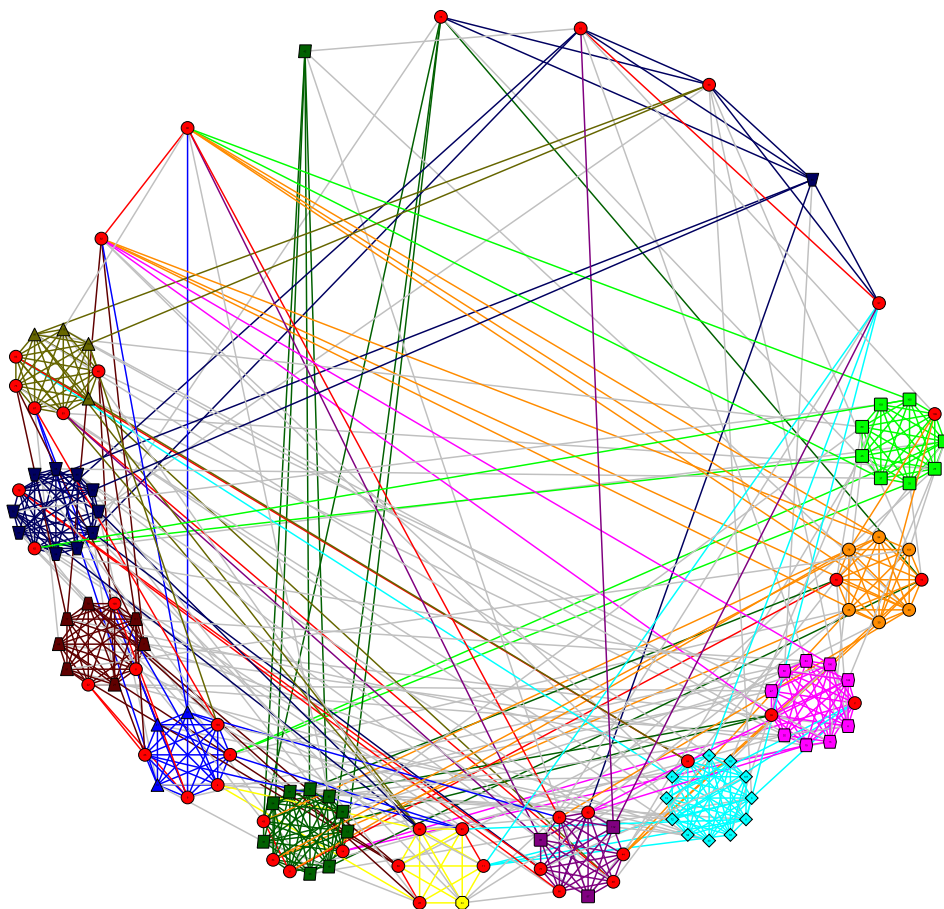
Appendix C. American College Football Network

Appendix C.1. Clique description

Clique percolation does not work with order three cliques, see figure C1. Only four communities are identified, two correspond to one conference each, one is based on the clique of four independents and all the remaining conference teams are all in one giant community.
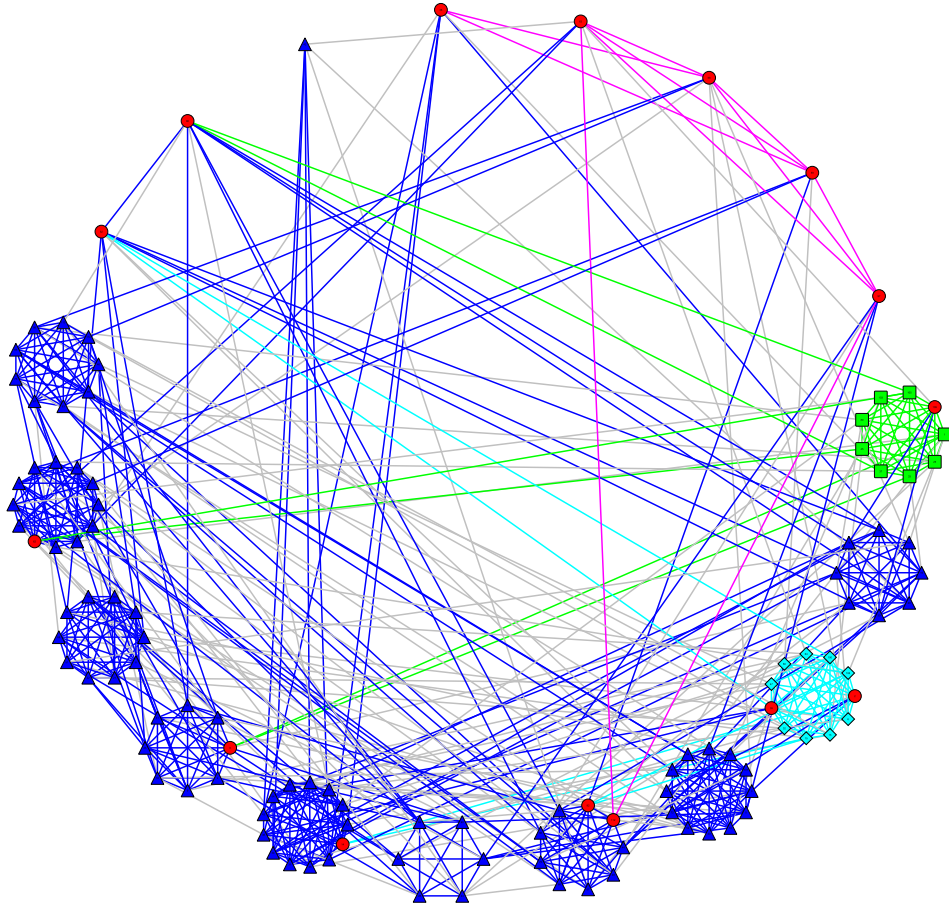
Figure C1. Network based on games played between teams of the Division 1-A American College Football league. The vertices are placed in the same locations as figure 5. The community structure detected by clique percolation of the order three cliques. Most of the teams are assigned to one single community and the conference structure remains largely hidden.

However vertex partitioning of the $C^{(3)}$ order three clique graph still works, see figure C2. About two-thirds of the conference teams are placed in a unique community containing only teams from their conference and perhaps some independents. For the other third, it is still true that the vast majority of triangles containing these conference teams contain only other teams in the same conference. That is it is easy to classify the overlap as weak and accurate conference identification remains simple.

Figure C2. Network based on games played between teams of the Division 1-A American College Football league. The vertices are placed in the same locations as figure 5. The community structure detected by vertex partitioning of the order three clique graph $C^{(3)}(G)$ clearly identifies the teams in each conference. About one-third of conference teams are members of communities containing teams from other conferences. However the majority of triangles containing conference teams contain only other teams in the same conference and so conference identification is simple.

Clique percolation is very successful if we work with order four cliques, see figure 5. All the conferences are detected as single groups except for one conference which is split into two communities. In that case the split reflects the fact that teams in this conference are also divided into two divisions. There is a community made from a order four clique of four independent teams: Middle Tennessee State, Louisiana Monroe, Louisiana Lafayette and Louisiana Tech. The first three joined the Sun Belt Conference the following year while the latter joined the Western Athletic Conference. Finally a few teams are part of more than one community. Usually such teams are members of far many more cliques in their community centred on their conference. One exception is the case of two independents which are only part of one community based on one of the conferences: Notre Dame and Navy are part of the Big East Conference.

The vertex partition of the $D^{(4)}$ order four clique graph ($\gamma = 1.0$) finds exactly the same community structure as clique percolation except it does not split the conferences, finding one community per conference, see figure C3.
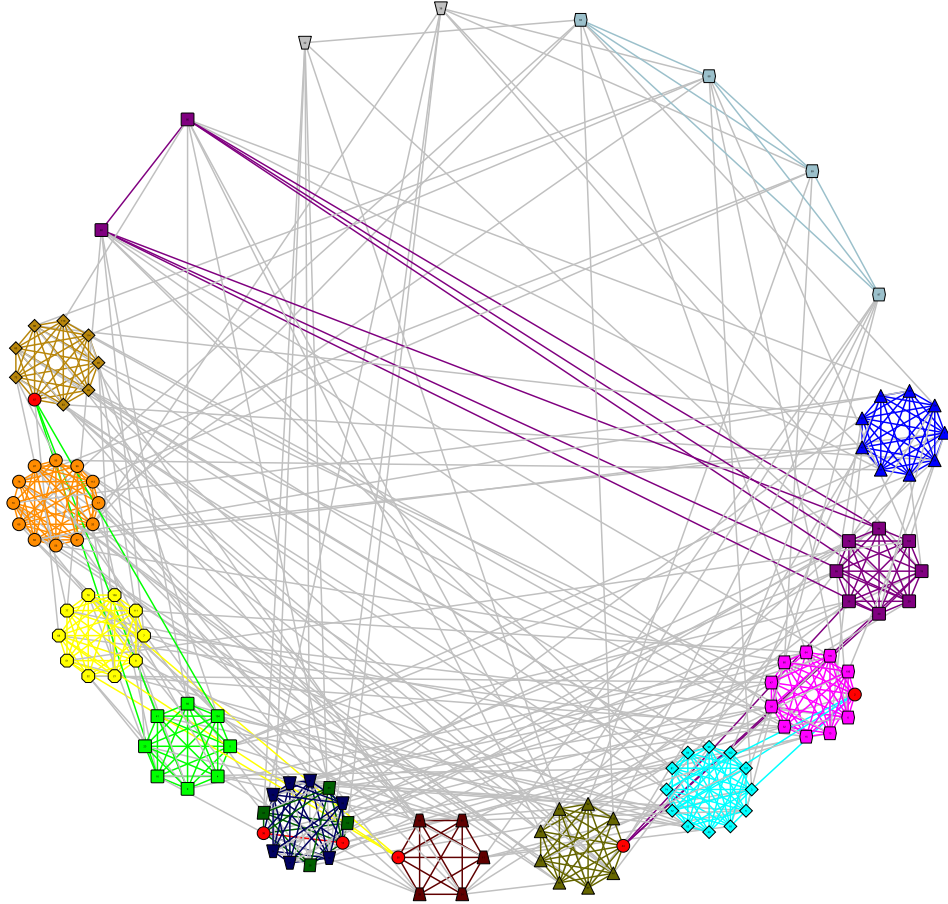
Figure C3. Network based on games played between teams of the Division 1-A American College Football league. The vertices are placed in the same locations as figure 5. The community structure is determined by optimising modularity for $\gamma = 1.0$ on the $D^{(4)}$ order four clique graph. The colours and shapes are chosen in the same manner as figure 5. The community structure detected using this vertex partition of the order four clique graph $D^{(4)}$ is identical to that found using order four clique percolation. It matches the conference structure almost perfectly.

The vertex partition of the $C^{(4)}$ order four clique graph ($\gamma = 1.0$) also a good match to the conference structure, see figure C4.
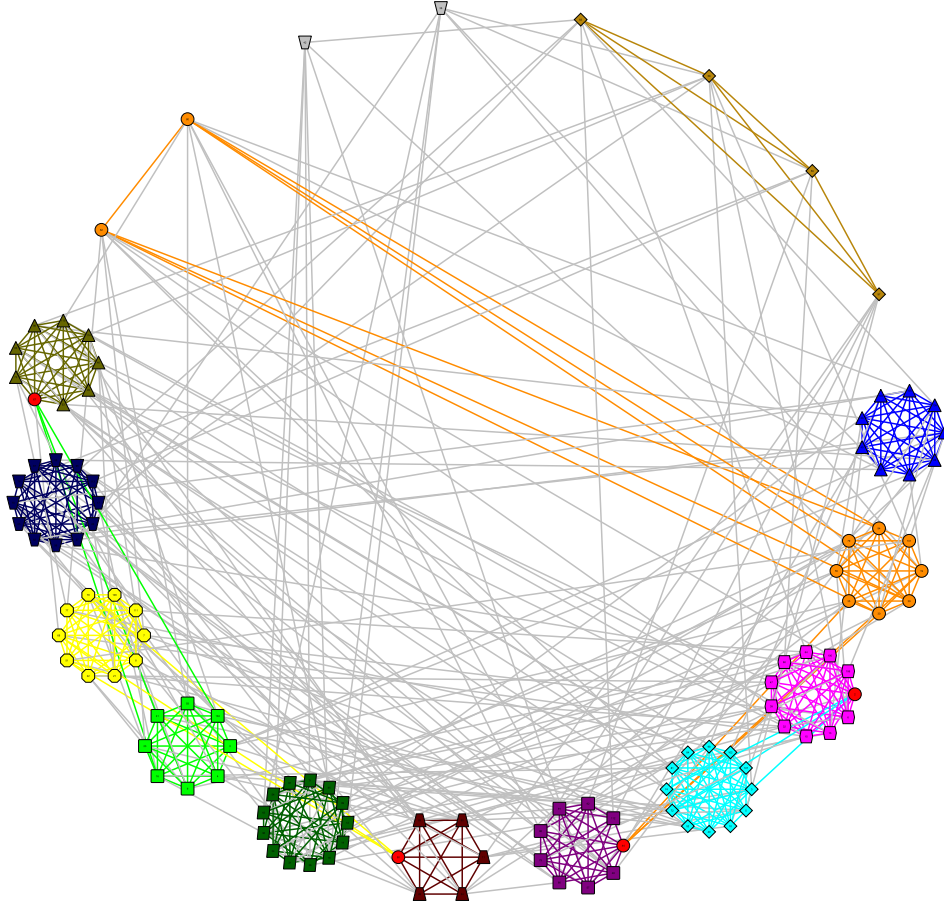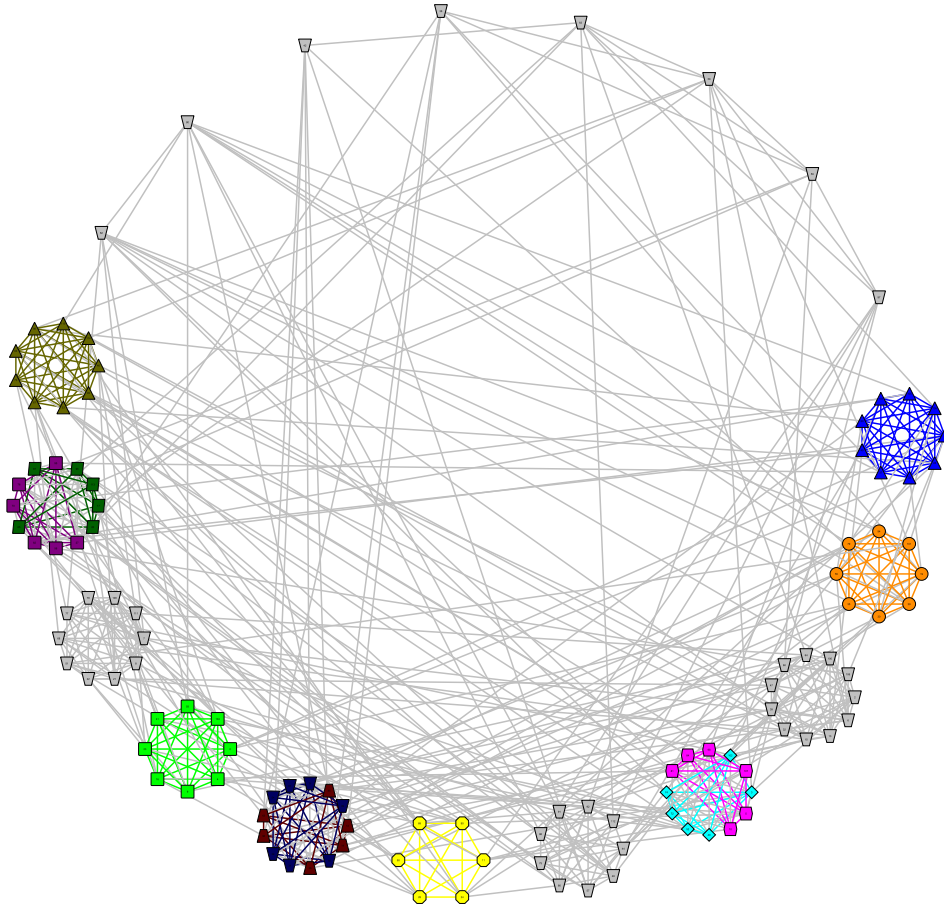

Figure C4. Network based on games played between teams of the Division 1-A American College Football league. The vertices are placed in the same locations as figure 5. The community structure is determined by optimising modularity for $\gamma = 1.0$ on the $C^{(4)}$ order four clique graph. The colours and shapes are chosen in the same manner as figure 5. The community structure detected using this vertex partition of the order four clique graph $D^{(4)}$ is almost identical to that found using order four clique percolation. The only difference is that the seventh conference (counting clockwise from the first), the Mid-American, is not split into two divisions. The community structure detected using this vertex partition of the order four clique graph $D^{(4)}$ matches the conference structure almost perfectly.

For order five clique percolation, we find 15 communities, each of which is entirely within one of the 11 conferences though one conference team, Alabama Birmingham in Conference USA, is now no longer part of a order five clique. No independent is in a order five clique. The method correctly splits two of the conferences into their two divisions but finds a single community for two other conferences with divisions. The one flaw is that it splits one other conference, which has two divisions, into three distinct communities. By way of comparison the vertex partition of the $C^{(5)}$ order five clique graph ($\gamma = 1.0$) gets exactly the same results except it does not split one conference

into three communities.

For order n cliques as we raise n from 6 upwards, too few teams are in cliques and an increasing number of conferences are not identified, see for example figure C5.

Figure C5. Network based on games played between teams of the Division 1-A American College Football league. The vertices are placed in the same locations as figure 5. The community structure is determined using 6-clique percolation. The colours and shapes are chosen in the same manner as figure 5. Now the three of the conferences are no longer detected though the substructure in three other conferences are detected.

Appendix C.2. Data Sources

The links between teams are derived from the file `football.gml` downloaded from Newman's web site (`http://www-personal.umich.edu/~mejn/netdata/`). This was compiled by Girvan and Newman [42] and gives a link for every Division I-A game of the American College Football league during Fall 2000. There is an edge for every game has been played between two teams.

However, there are two issues with the original file. First three teams met twice in one season, typically because of conference finals. The pairs were: teams 3 (Kansas State) and 84 (Oklahoma), teams 99 (Marshall) and 14 (Western Michigan), and teams 27 (Florida) and 17 (Auburn). For this work only one edge has been created for these three pairs and the graph is simple.

Secondly, the assignments to made to conferences (the `value` tag in the original file) seems to be for the 2001 season and not the 2000 season. In particular the Big West conference existed for football till 2000 while the Sun Belt conference was only started in 2001. The new conference assignments used in this paper are given in table C1.

Index	Conference	Index	Independent
0	Atlantic Coast	11	Notre Dame
1	Big East	12	Navy
2	Big Ten	13	Connecticut
3	Big Twelve	14	Central Florida
4	Conference USA	15	Middle Tennessee State
5	Big West	16	Louisiana Tech
6	Mid-American	17	Louisiana Monroe
7	Mountain West	18	Louisiana Lafayette
8	Pacific Ten		
9	Southeastern		
10	Western Athletic		

Table C1. Table listing the Division 1-A American College Football conferences for the Fall 2000 season. The indices refer to those used in this paper. In the figures showing the network based on games played in the 2000 season, teams from each conference are placed in order clockwise in a large circle with conference 0 (Atlantic Coast) at 3 o'clock and the last conference (10, Western Athletic) just after 9 o'clock.

Using the conference assignments given in table C1 for the 2000 season means that corrections were made to the values given in the file `football.gml` downloaded from Newman's web site for Western Athletic and Independent teams. In addition the following changes were needed:-

- N.Texas (v11) is in conf 5 Big West (not Sun Belt, GN conference 10)
- Arkansas State (v24) is in conf 5 Big West (not Sun Belt GN conference 10)
- Boise State (v28) is in conf 5 Big West (not Western Athletic GN conference 11)
- Idaho (v50) is in conf 5 Big West (not Sun Belt, GN conference 10)
- Louisiana Tech (v58) is in conf 16, an Independent (not Western Athletic GN conference 11)
- Louisiana Monroe (v59) is in conf 17, an Independent (not Sun Belt GN conference 10)
- Middle Tennessee State (v63) is in conf 15, an Independent (not Sun Belt GN conference 10)
- New Mexico State (v69) is in conf 5 Big West (not Sun Belt, GN conference 10)
- Utah State (v90) is in conf 5 Big West (not Independents, GN conference 5)
- Louisiana Lafayette (v97) is in conf 18, an Independent (not Sun Belt GN conference 10)
- Texas Christian (v110) is in Western Athletic conf 10 (not Conference USA GN conference 4)

A sample of the games have been checked using the results given on the "College Football Data Warehouse" (`www.cfbdatawarehouse.com`) and using various Wikipedia entries for the different conferences and teams.

Appendix D. Further Benchmark example

Below are some more examples of communities found on the benchmark graph.
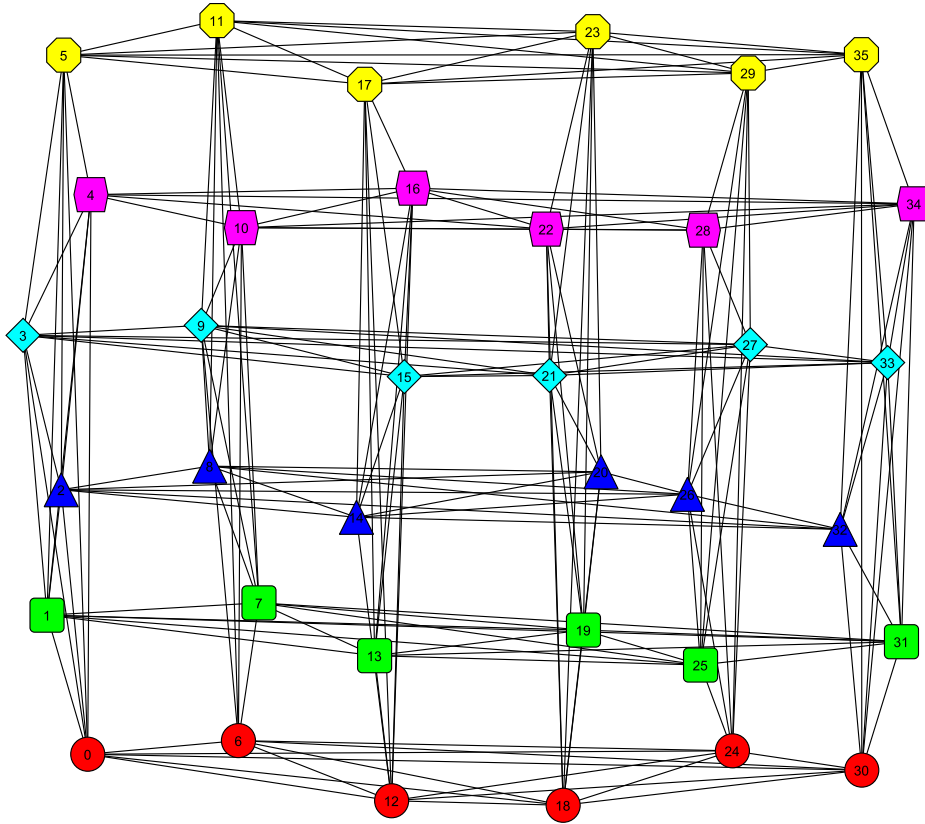
Figure D1. A vertex partition of the benchmark graph with 216 triangles using the Louvain algorithm to maximise modularity with $\gamma = 1.0$. On this run, the six communities associated with the columns are found, indicated by the vertex shapes and colours, but the row communities are completely missed.
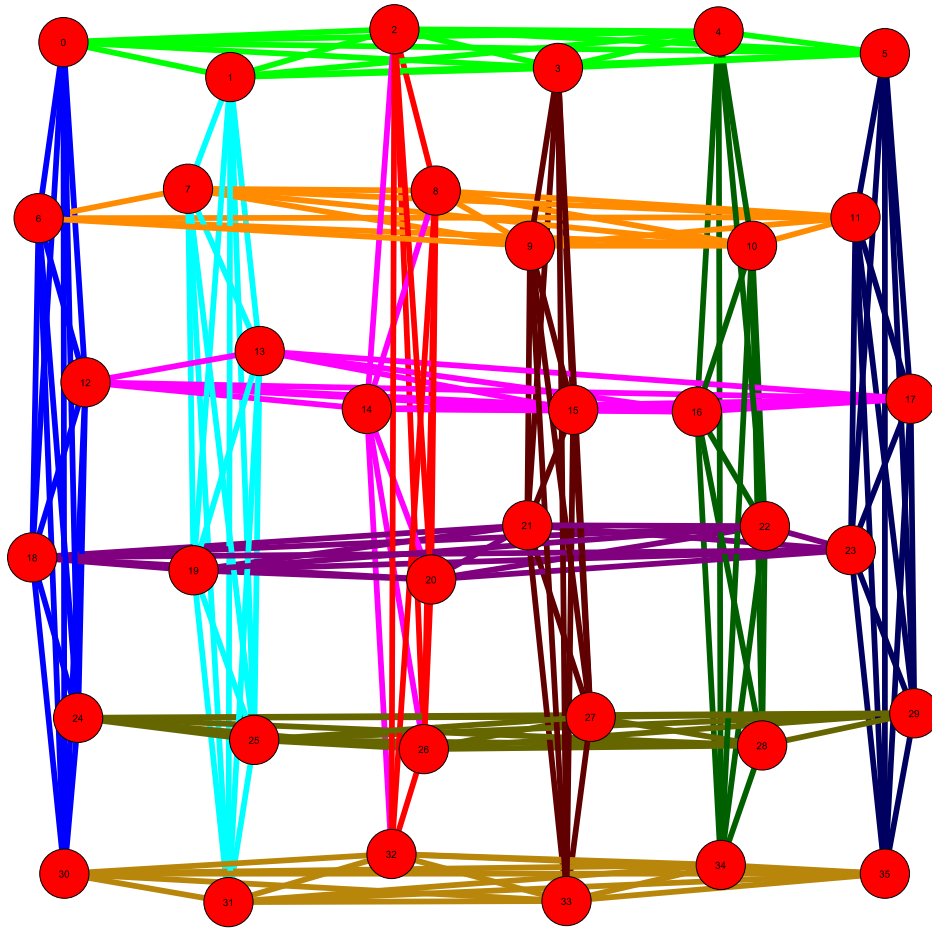
Figure D2. The community structure based on the partition of order three cliques of the same benchmark graph as in figure D1. Produced by applying the Louvain algorithm to the $D^{(3)}$ clique graph, maximise modularity with $\gamma = 2.5$. Twelve communities were produced on this run, matching the column and row communities perfectly. This is indicated by the edges in each row having a unique colour, and similar for the columns. The one flaw is that the edges in the third column are assigned to two communities, one unique to that column, and the other used for the third row from the top.

The community structure based on the partition of order three cliques of the same benchmark graph as in figure D1. Produced by applying the Louvain algorithm to the $D^{(3)}$ clique graph, maximise modularity with $\gamma = 2.5$. Twelve communities were produced on this run, matching the column and row communities perfectly. This is indicated by the edges in each row having a unique colour, and similar for the columns. The one flaw is that the edges in the third column are assigned to two communities, one unique to that column, and the other used for the third row from the top.
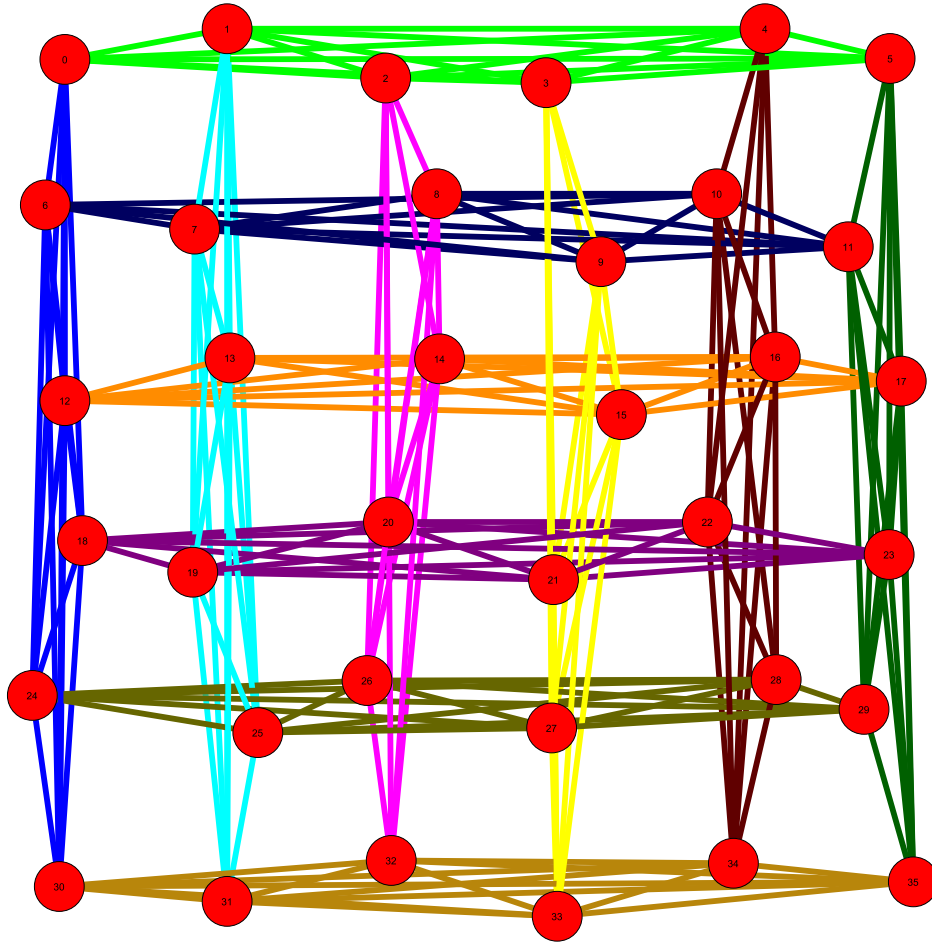
Figure D3. The community structure based on the partition of 3-cliques of the same benchmark graph as in figure D1. Produced by applying the Louvain algorithm to the $C^{(3)}$ clique graph, maximise modularity with $\gamma = 3.0$. Twelve communities associated with the columns are found matching the column and row communities perfectly. This is indicated by the edges in each row having a unique colour, and similar for the columns.

The community structure based on the partition of 3-cliques of the same benchmark graph as in figure D1. Produced by applying the Louvain algorithm to the $C^{(3)}$ clique graph, maximise modularity with $\gamma = 3.0$. Twelve communities associated with the columns are found matching the column and row communities perfectly. This is indicated by the edges in each row having a unique colour, and similar for the columns.
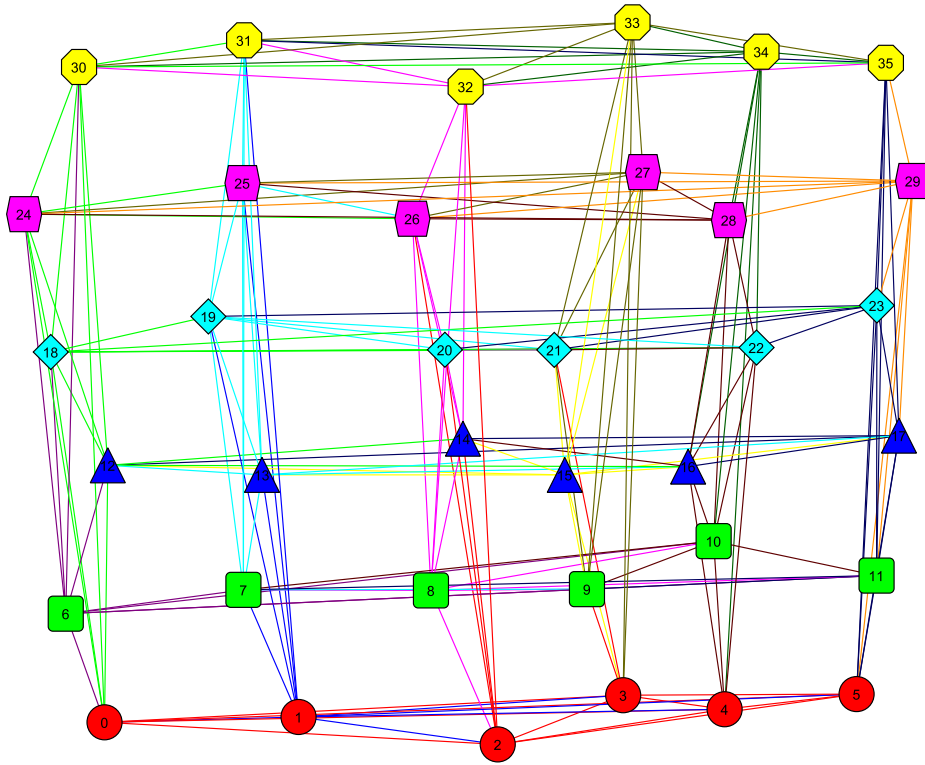
Figure D4. A vertex partition and an edge partition of the same benchmark graph as used in figure D1. The edge partition is found by finding a vertex partition of the weighted line graph $D(G)$ described in [39]. Both partitions are found using the Louvain algorithm to maximise modularity (9) with $\gamma = 1.2$. On this run, the vertex partition finds the six communities associated with the rows, indicated by the vertex shapes and colours, but the row communities are completely missed. The edge partition find the six columns and one of the row communities, as indicated by the edge colours.

A vertex partition and an edge partition of the same benchmark graph as used in figure D1. The edge partition is found by finding a vertex partition of the weighted line graph $D(G)$ described in [39]. Both partitions are found using the Louvain algorithm to maximise modularity (9) with $\gamma = 1.2$. On this run, the vertex partition finds the six communities associated with the rows, indicated by the vertex shapes and colours, but the row communities are completely missed. The edge partition find the six columns and one of the row communities, as indicated by the edge colours.