Sports Coaching Networks: Using Community Detection to Analyse Coaching Strategies

Candidate Number 654352

March 24, 2014

Abstract

We construct a directed social network of coaches based on tutelage, from which we examine community structure based upon connections in the network given by community-detection algorithms. We use this approach to investigate communities of coaches from the National Football League for American football and Major League Baseball, both sports leagues in the United States. We analyse patterns amongst communities of head coaches from different types of networks and investigate whether the history of who a coach has worked for impacts on how successful they are likely to become. Communities can also be formed based upon different strategies employed by coaches, which are studied in relation to the communities found from algorithms.

Contents

1 Introduction

The study of networks is a rapidly growing area of research with increasingly many applications, including the study of the world of sports. The areas studied within sports are varied, from methods on how to rank teams [7], to analysing match-ups between the pitcher and the batter in baseball [37]. We are interested in the lineages of coaches and patterns between them. This has previously been investigated by Fast and Jensen [9] using probabilistic models, but we hope to take a networks-oriented approach to this.

When analyzing how certain tactics have evolved in sports, analysts often consider 'coaching trees' (local groups of coaching networks) to assess how schemes have been passed from coach to coach [15]. For example, it is often remarked that in American football, a tactical scheme called the 'West Coast Offense' was developed by Paul Brown and perfected by Bill Walsh, a disciple of his, and has since been employed by many successful coaches who used to work for them [9].

Community detection in networks is often used to determine underlying patterns by grouping related entities together. We loosely consider a *community* to be a collection of entities more closely connected to each other than to others outside of the community [10, 30] (for a more rigorous definition see Section 2.8). By applying a variety of community-detection algorithms, we can establish reasonable partitions of the network into communities [21]. This has previously been done in a variety of settings [33], for example in legislative networks [32], using data on voting similarity and in communication networks [31] from mobile phone usage data.

In this dissertation, we study networks of coaches and apply community-detection algorithms [4, 24] to determine suitable communities in which different coaches can be placed. We study whether the notion of communities is realistic in this setting, and if it is we examine whether 'successful' coaches are more likely to be grouped together than 'unsuccessful' coaches. This could imply that a particular coach is more or less likely to be successful based purely on who they are 'similar' to in the network. This could be useful to predict the success of current coaches who are about to become head coaches but who have no experience at that job.

2 Network Concepts and Definitions

In this section we will introduce important network concepts and definitions.

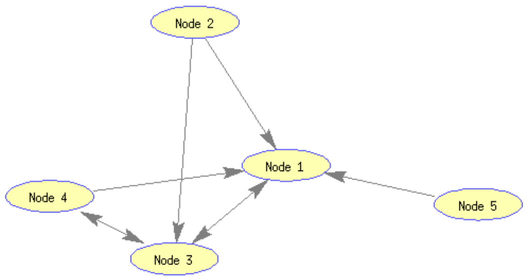
2.1 Definition of a Network

We define a *network* to be a system of interconnected entities (*nodes*) and their interactions (*edges*), with roots linked to graph theory [29].

The simplest type of network has one type of node and one type of edge, with uniform edge weights and without a notion of direction. We call this simple network construction a *graph*. Mathematically, a graph is a set of nodes V, where each node $v \in V$ represents a member of the network, in addition to a set of edges $E \subseteq V \times V$ that describes the relationships between these nodes. We denote an edge from i to j by (i, j). As our graph is without direction, the existence of (i, j) implies the existence of (j, i). The ends of an edge are called *stubs*. The network is given by the sets of nodes and edges $G = (V, E)$, which can be described by an adjacency matrix (see Section 2.4). Throughout this dissertation we shall use the notation for the total number of nodes N and the number of edges m.

2.2 Directed and Undirected Networks

Many networks include a directionality attached to edges between nodes. In particular, all of the data for coaching lineages will be directional. We define a *directed network* to be a network G where the relationship between one node and another is not necessarily reciprocated (for example see Figure 1). This means the presence of the edge (i, j) does not imply the existence of (j, i). If edges are reciprocated, we have an *undirected network*.



(a) The directed network

$$\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

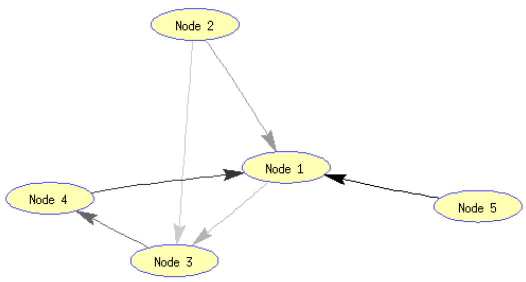
(b) The corresponding adjacency matrix

Figure 1: A directed network.

2.3 Weighted Networks

We define a *weighted network* to be a network where edges have weights associated with them. A weighted network is a collection of nodes V along with edges $E \subseteq V \times V$ in addition to a set of weights W defined on each edge such that $W(E) \subset \mathbb{R}$. This gives a weighted network $G = (V, E, W)$. A network without weights attached to nodes is called a *binary* or an *unweighted* network. The *topology* of a weighted network refers its binary counterpart $G = (V, E)$. For an example of a weighted network see Figure 2.

Edge weights can represent importance of an edge in the network; for example, in a friendship network the edge weight may represent the length of time spent as friends. Both binary and weighted networks will be considered in this dissertation.



(a) The weighted network

$$\begin{pmatrix} 0 & 2 & 0 & 4 & 5 \\ 0 & 0 & 0 & 0 & 0 \\ 1.67 & 1 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(b) The corresponding adjacency matrix

Figure 2: The weighted network, with larger edge weights represented by darker arrows.

2.4 Adjacency Matrices

One of the most common ways to represent a network mathematically is with an *adjacency matrix* [29]. If we arbitrarily label the nodes of our matrix with integers $1, \ldots, N$; then we define the $N \times N$ adjacency matrix $\mathbf{A}$ with elements A_{ij} such that

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge from } j \text{ to } i, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In a directed network (for example in Figure 1b), the adjacency matrix is asymmetric in general.

This definition can be extended to a weighted network, so that if an edge exists from j to i then the corresponding entry A_{ij} is given by the weight associated with that edge, as shown in Figure 2b.

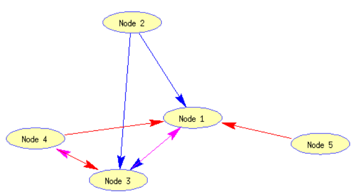
2.5 Multiplex Networks

A *multiplex network* [18] is a generalisation of a standard network, in which different types of edges that exist between nodes. We have a set of nodes V, and a collection of sets of edges $E_1, E_2, \ldots, E_S \in V \times V$, which define the network $G_{mul} = (V, E)$ where $E = \{E_1, E_2, \ldots, E_S\}$. We can consider a multiplex network as S different *layers* of the same nodes with each layer having a single edge type.

The adjacency tensor for a multiplex network is given by the $(N \times N \times S)$ tensor $\mathbf{A}$ such that the elements A_{ijs} satisfy

$$A_{ijs} = \begin{cases} 1, & \text{if there is an edge from node } j \text{ to node } i \text{ with edge type } s, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

For example see Figure 3.



(a) The multiplex network

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(b) Layer of adjacency tensor for edge type 1

$$\begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(c) Layer of adjacency tensor for edge type 2

Figure 3: A directed multiplex network with 2 types of node, which are represented by red and blue, respectively. Two nodes can share multiple edge types, which we represent with magenta edges.

2.6 Temporal Networks

To examine how a network develops over time we consider a *temporal network* [14]. In our context, we are only concerned with discrete-time networks, which we represent as a set of nodes V and a collection of sets of edges $E_1, E_2, \ldots, E_T \in V \times V$, which refer to the presence of edges at different times $t = 1, \ldots, T$ of the development of the network. If we let $E = \{E_1, E_2, \ldots, E_T\}$ then we define the network $G_{tem} = (V, E)$. This definition using multiple layers is similar to the multiplex network; networks of this type are known as *multilayer networks* [18].

The adjacency tensor is given by the $N \times N \times T$ tensor $\mathbf{A}$ such that the elements A_{ijt} satisfy

$$A_{ijt} = \begin{cases} 1, & \text{if there is an edge between node } j \text{ and node } i \text{ at time } t, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

We will consider temporal networks for the coaching networks, to give an added time dimension due to the change in coaching teams from year-to-year.

2.7 Connectedness and Components

Another important property to consider is whether nodes are connected to each other. We define a *path* from node a to node b in a network to be a sequence of nodes $\{v_1, v_2, \ldots, v_p\}$ with $v_1 = a$ and $v_p = b$ such that there is an edge from node v_{i-1} to node v_i for $i \in \{1, \ldots, p-1\}$. An undirected network is said to be *connected* if there exists a path between every pair of nodes $a, b \in V$ and is called *disconnected* if there exists nodes $a, b \in V$ such that there is no path from a to b.

We extend this definition to a directed network, which we call *weakly connected* if the network is connected edge directionality is ignored. There are other notions of connectedness in directed networks [29] but these are not necessary for this dissertation.

This definition can be extended for multilayer networks by considering a *network aggregation* of the networks [18], where we form a matrix $\tilde{\mathbf{A}}$ from the tensor $\mathbf{A}$ so that $\tilde{A}_{ij} = \sum_{r=1}^{S} A_{ijr}$ and then consider whether the weighted network with adjacency matrix $\tilde{\mathbf{A}}$ is weakly connected.

We also define a *connected component* in an undirected network as a subset of nodes such that the network formed by these nodes and the edges between them is connected. This can be extended for a directed network to a *weakly connected component* which is a subset of nodes which are weakly connected. In multilayer networks we apply *network aggregation* as suggested by Kivelä et al. [18], to find connected components. We shall assume in this dissertation that connected components are *maximal*, so no additional nodes can be added to a component whilst maintaining connectedness.

2.8 Communities

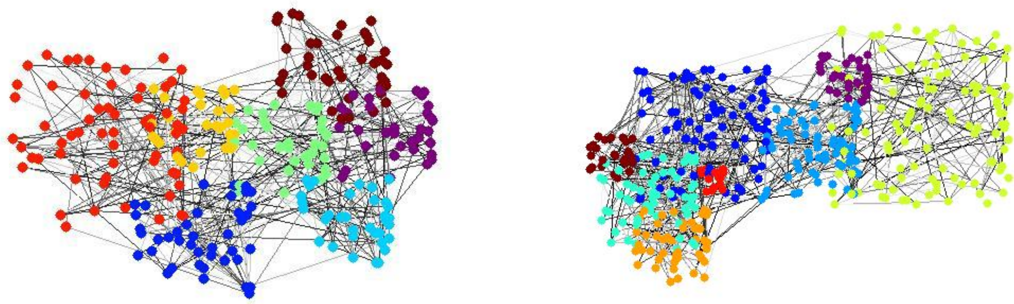
It is often useful to partition a network into *communities*. We shall consider a community to be a grouping of nodes with relatively high edge density to other nodes in the same community, but with low edge density to nodes in other communities [33]. Communities are an example of *mesoscopic structure* which lies between *macroscopic* (whole world) and *microscopic* (individual node) structures.

As the definition of a community is not rigorous, we cannot talk about the existence of 'optimal communities'. Instead we often look to optimise an *objective function* which attempts to describe what an optimal partition is. From this we can formulate an idea of a reasonable partition into communities by considering *community-detection algorithms* (see Section 5).

Community structure is able to reveal mesoscale network features that are not immediately obvious from the data. For example, using data obtained by a Zachary study of the social network within a karate club which split into two karate clubs [41], community-detection algorithms can predict how the split was going to occur based upon only friendship data [33].

3 The Sports Networks

In this dissertation, we are interested in lineage networks for sports coaches. This will be similar to work done on mathematical lineage networks by Malmgren [22], which looks at the performance of students in mathematics departments by considering correlations in the number of protégés that a mentor trains (fecundity).



(a) The NFL Network (b) The MLB Network

Figure 4: Diagrams of the binary forms of the two networks studied. Colour schemes are based upon the communities found in Section 6 for binary directed networks. Communities are placed from the Fruchterman-Reingold Algorithm [12] with the Kamada-Kawai Algorithm [16] to place nodes within the community.

Lineage networks have a directed edge from node A to node B if node B acted as a protégé under node A. For our networks, the head coaches are the mentors and all assistant coaches are protégés. We only consider situations in which B goes on to become a head coach, so all nodes in the network have been a head coach.

So that the mean degree (see Section 4.1) of the network is large enough to give large connected components, we sought networks that have large coaching teams where it is common that future head coaches are found from the ranks of other coaching teams. The two networks that we studied were the National Football League [1, 2] (NFL, a league for American football) and Major League Baseball [3] (MLB, a league for baseball), which are both professional sports leagues in the United States. Note that in baseball the head coach is usually called the 'manager', which is how we shall refer to the head coach when discussing baseball.

For both MLB and the NFL, we shall only consider the largest weakly connected component of coaches. In the NFL, this includes all 279 coaches for which we have data; in MLB, however, our data for managers goes back to an era before large coaching teams became the norm. This has narrowed the pool of potential managers to study to 443 of 697 managers. The managers analysed include all managers except for 4 who have coached in the last 70 years. In the early days of baseball, only small coaching teams existed, so there are fewer edges between coaches from this era.

3.1 Types of Network Analysed

We use data for the MLB going back to 1871 and for the NFL going back to 1957. t For MLB and the NFL, we will analyse different types of networks. We use data for the MLB going back to 1871 and for the NFL going back to 1957. The types of network studied are:

- A binary directed network, where there is an edge (i, j) if coach j was an assistant under coach i.
- A weighted directed network, where weights are given by the number of years (rounded up to the nearest integer) that coach B coached under coach A. We consider both weighted and binary networks to examine whether multiple years of tutelage makes a significant difference to network community structure compared to ignoring this information.

- A temporal network, in which each layer in the network is given by the coaching teams present at that point in time. In MLB, due to computational constraints of the large matrices involved in calculations, only the past 50 years were considered. This gives a weakly connected component of 309 connected coaches.

- A multiplex network for MLB, with 2 layers. The first layer is the binary directed coaching network from above. The second layer includes playing data for coaches who are ex-players. In the additional layer, there is an edge (i, j) if coach j played under coach i. This gives a largest connected component of 692 managers. This is practical for MLB as a majority of coaches are ex-MLB players whilst in the NFL this is not as common (there are 8 former players who currently have jobs as head coaches [39], but 4 of those were career backups as players). This can be compared with the results from the single-layer coaching network to examine whether who a player plays under influences his coaching.

- A network of 210 'defensive leaders' for the NFL. In the NFL, assistant coaches tend to work exclusively on either offense or defence, with head coaches either being defensively- or offensively-minded. A head coach is considered to be defensively-minded if he ever worked as an assistant coach in a defensive role. We form edges between defensive coaches to either the head coach, if he was defensively-minded, or else to the defensive coordinator, in a similar fashion to the standard head coach matrix. This is interesting to examine in combination with data on defensive formations (see Section 7.2).

I did the cleaning of all data given above in addition to any results data. Details on the processes involved are given in Appendix B.

4 Basic Properties for the Coaching Networks

In this section we present a variety of ways for analysing networks- both from general networks theory and also some measures relevant only to the coaching networks- which gives an idea of the features of the networks, as well as to analyse community structure. Many of these properties are applicable to only the single-layer networks, but we shall state where there are extensions to multilayer networks.

4.1 Network Size, Number of Edges, Weights

The simplest diagnostics for a network with adjacency matrix $\mathbf{A}$ are given by number of nodes N and total number of edges $m = \sum_{i,j} A_{ij}$. The *in-degree* of node j is given by $k_j^{\text{in}} = \sum_{i=1}^{N} A_{ij}$, giving the total number of edges going into j [29]. *Out-degree* can be similarly defined as the number of edges going out of a node i, given by $k_i^{\text{out}} = \sum_{j=1}^{N} A_{ij}$. This gives *mean degree*

$$c = \sum_{j=1}^{N} k_j^{\text{in}} = \sum_{i=1}^{N} k_i^{\text{out}} = \frac{m}{N}. \tag{4}$$

These definitions can be extended to a weighted network where the same calculations apply; here we will have *total edge strength* $m = \sum_{i,j} A_{ij}$ and *mean degree strength* $c = \frac{m}{N}$.

For the multilayer networks, we have to be more careful with how to generalise these definitions. Total nodes N is the total number of nodes, not double-counting nodes that are in multiple layers. To calculate total edges and degree, we apply network aggregation (see Section2.7), to form a matrix $\tilde{\mathbf{A}}$

	NFL				MLB			
	Binary	Weight	Temporal	Defense	Binary	Weight	Temporal	Multiplex
Nodes	279	279	279	210	443	443	309	692
Layers	1	1	56	1	1	1	50	2
Edges	1003	2992	2992	517	1617	3749	2710	5987
Mean Degree	3.65	8.46	8.46	2.46	3.61	10.8	8.77	8.66

Table 1: Basic diagnostics of the largest weakly connected component of the coaching networks. Edges refers to either total edges (binary) or total edge strength (weighted). Many statistics are the same for the NFL weighted and temporal networks as the weighted network is an aggregation of the temporal network but not for MLB as the temporal network only goes back 50 years.

from the tensor $\mathbf{A}$ so that $\tilde{A}_{ij} = \sum_{r=1}^{S} A_{ijr}$. This gives a weighted network with adjacency matrix $\tilde{\mathbf{A}}$ from which we apply the methods as given above.

Note that for coaching networks we can interpret mean degree as the mean number of other coaches that a head coach has worked under, and mean degree strength as the mean number of years a head coach has worked as an assistant.

4.2 Win Fraction and 'Coach Influence' Diagnostic

To look at the success of teams coached by a head coach, we use a quantity known as *win fraction*, used in many American sports, which is simply calculated for our purposes as

$$W = \frac{\text{games won} + 0.5 * \text{games tied}}{\text{total games played}}. \tag{5}$$

When discussed in American sports, the phrase *win percentage* is more commonly used, but since it is a number between 0 and 1, we shall use win fraction.

To compare between communities, I have developed a very simple quantitative diagnostic for a coaches influence called *coach influence (CI)*. This is not intended to be a complex measure to exactly rank each coach; but to give a vague idea of which communities contain more successful coaches than others. It is intended to be a reasonable measure when aggregated over many coaches.

To calculate coach influence, we consider 3 quantities of importance for the influence of a coach to the development of strategy in different sports:

- Tenure, which is the number of years as a head coach.
- Out-degree, or the number of a coaches' assistants that went on to become head coaches.
- Win fraction.

For a given coach, if they have the T^{th} shortest tenure, the D^{th} smallest out degree and W^{th} lowest win fraction then the un-averaged coach influence CI_u for the given coach is given by

$$CI_u = T + D + W, \tag{6}$$

so that larger scores represent more 'influential' coaches. To give a mean CI of 1 amongst coaches, we consider the mean coach influence score, CI_{avg} to give coach influence $CI = \frac{CI_u}{CI_{avg}}$ for a coach.

As an example, in the NFL the diagnostic ranked Don Shula the highest out of coaches, which is reasonable given he coached for 33 years winning 67% of his games and won two Superbowls [34]. For

	NFL		MLB	
Rank	Coach	Score	Coach	Score
1	Don Shula	11.3	Joe McCarthy	9.3
2	Marty Schottenheimer	6.4	Davey Johnson	6.2
3	George Allen	5.8	Walter Alston	6.2
4	Bill Belichick	5.8	John McGraw	5.9
5	Tom Landry	5.0	Bobby Cox	5.7

Table 2: Top 5 coach influence scores in the NFL and MLB.

the top five coaches in NFL and MLB see Table 2, although this is intended to be a measure to be aggregated amongst coaches within a community and not an absolute ranking of individual coaches.

5 Modularity Optimisation

For a directed network there are many different ways that one can attempt to find communities. Most methods involve the following steps [21]:

1. Define an objective function which gives a rating of the partition into communities.
2. Use an algorithmic techniques to optimise this function.

Many methods involve the optimisation of an objective function known as *modularity* [30]. The optimisation of modularity can take place in a variety of ways, including greedy algorithms [26], simulated annealing [35], and spectral methods [20] (which we use in this dissertation). Many of these algorithms were originally designed for undirected networks but can be extended to directed networks [21].

All of the code to calculate modularity and detect communities was written by me. For some sample code for coding modularity and the Newman-Leicht algorithm, see Appendix C.

5.1 Definition of the Objective Function

The modularity function that we use is the directed extension given by Arenas et al. [4] to the original method of Newman and Girvan [30]. We shall present both here as the modularity for undirected networks helps explain it for directed networks.

For an undirected network with c_i representing the community containing node i and node degree $k_i = \sum_{j=1}^{n} A_{ij}$, we define the *modularity*

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j), \tag{7}$$

where δ is the Kronecker delta function

$$\delta(c_i, c_j) = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{ lie in the same community,} \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

This gives a value for modularity that is effectively the fraction of edges that connect nodes within the same community minus the expected fraction of edges to connect nodes inside the community [30], based on the *configuration model* [10] as the null model. Values lie between $-1/2$ and 1 [6]. Larger values for modularity are a desirable trait for partitions as this shows that the expected fraction of edges that remain within a community is higher than would be expected from the null model.

Modularity can be used to quantify the quality of a given network partition (from communities based upon ‘ground-truth’ data or from another method) or it can be used as a function which is to be optimised in order to find communities. The main disadvantage of using it as an optimisation tool is that to find the global maximum of modularity is an NP-hard problem [6]. On top of this many partitions lie at locally maximum values of modularity which are not similar to the partitions associated with the global maximum. This leads to maximising modularity becoming computationally impossible for even medium sized networks, with communities found not necessarily being reasonable estimations of partition generated from maximum modularity.

It should be noted that there has been criticism on the use of modularity as a quality function. Good et al. [13] have pointed out that modularity scales across both network size and density, so it is often not practical to compare modularity scores across different networks to examine which has the ‘best’ partition.

Another criticism of modularity optimisation is that in many situations, it may be more appropriate if nodes could be members of multiple communities, for example in a social network a person can be a member of many social groups.

For the purposes of this dissertation, we use modularity due to its simplicity and how easy it is to use. It is also a common measure of the quality of a partition used by many other authors (for example see [10, 20, 40]). However, it should be kept in mind if a coach is placed in a community by the algorithm, then other communities may also be suitable for that coach. We should also remember the partition given by an algorithm may not be similar to the partition which maximises modularity.

One can also generalise modularity optimisation for directed networks. The observation made by Arenas et al. [4] was that for the null model, the presence of a directed edge from node i to node j depends on the the out-degree of i and the in-degree of j. If i has high out-degree and low in-degree whilst j has low out-degree and high in-degree, we would be more likely to find the edge (i, j) than (j, i).

We may define the *directed modularity function*

$$Q = \frac{1}{m} \sum_{i,j} \left(A_{ij} - \frac{k_i^{\text{out}} k_j^{\text{in}}}{m} \right) \delta(c_i, c_j). \tag{9}$$

This is defined so that the modularity is again given by the fraction of edges that fall within communities minus the expected fraction of edges; similar to the undirected function but now using a generalisation to the configuration model for a directed network [29] as the new null model. As the generalisation of modularity is only a minor change in the definition, the same criticisms apply as the undirected definition.

5.2 Optimisation of the Modularity Using Spectral Methods

As maximising modularity is an NP-hard problem [6], we need to use an algorithm to find communities which gives a close approximation to the maximum value of modularity whilst using little computational time. For this dissertation, we shall use the Newman-Leicht spectral method [20] which is generalised from the algorithm of Newman [27], but there are many other possibilities which could be utilised (see [21, 26, 35]).

For the Newman-Leicht method, we first consider the problem of separating our directed network into two communities. We define s_i to be $+1$ if node i is in community α and -1 if it is in community

β, so that $N = \sum_{i=1}^{n} s_i^2$ and $\delta(c_i, c_j) = \frac{1}{2}(s_i s_j + 1)$. It follows from the definition of modularity that

$$\begin{aligned} Q &= \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i^{\text{out}} k_j^{\text{in}}}{m} \right) \frac{1}{2}(s_i s_j + 1) \\ &= \frac{1}{2m} \sum_{i,j} s_i B_{ij} s_j \\ &= \frac{1}{2m} \mathbf{s}^{\text{T}} \mathbf{B} \mathbf{s}, \end{aligned} \tag{10}$$

where we define the vector $\mathbf{s}$ to be the vector with elements s_i and the *modularity matrix* $\mathbf{B}$ with elements $B_{ij} = A_{ij} - \frac{k_i^{\text{out}} k_j^{\text{in}}}{m}$. Note that we use $m = \sum_{i,j} A_{ij} = \sum_i k_i^{\text{out}} = \sum_j k_j^{\text{in}}$. The Newman-Leicht algorithm seeks to find $\mathbf{s}$ to maximise modularity Q.

Spectral methods are eigenvector-based methods which require real entries for the leading eigenvector. This cannot be guaranteed for general asymmetric matrices, so we need to symmetrise our modularity by adding (10) to its transpose $Q = \frac{1}{2m} \mathbf{s}^{\text{T}} \mathbf{B}^{\text{T}} \mathbf{s}$ to give

$$Q = \frac{1}{4m} \mathbf{s}^{\text{T}} (\mathbf{B} + \mathbf{B}^{\text{T}}) \mathbf{s}. \tag{11}$$

It is important to note that there is a distinction between symmetrising the matrix at this stage compared to symmetrising the adjacency matrix and applying the method for undirected networks. In particular $\mathbf{B} + \mathbf{B}^{\text{T}}$ is not equal to the modularity matrix we would have found if we had started with $\mathbf{A} + \mathbf{A}^{\text{T}}$. For general community-detection algorithms, if we are applying a single community to each node rather than an in-community and an out-community, it is necessary to have a form of symmetrisation at some point [21].

We now write the vector $\mathbf{s}$ as a linear combination of the orthonormal eigenvectors $\mathbf{v_i}$ with eigenvalues β_i labelled in descending order size $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_N$ for the matrix $\mathbf{B} + \mathbf{B}^{\text{T}}$. We write $\mathbf{s} = \sum_i a_i \mathbf{v_i}$, as N orthonormal eigenvectors span $\mathbb{R}^{\text{N}}$, with $a_i = \mathbf{v_i}^{\text{T}} \mathbf{s}$. We ignore the factor of $\frac{1}{4m}$ as it does not affect the partitions obtained, so that

$$\begin{aligned} Q &= (\sum_i a_i \mathbf{v_i^T})(\mathbf{B} + \mathbf{B}^T)(\sum_i a_i \mathbf{v_i}) \\ &= \sum_i \beta_i (\mathbf{v_i^T s})^2. \end{aligned} \tag{12}$$

For normalised $\mathbf{s}$, a good approximation to the maximum of Q is to take $\mathbf{s}$ parallel to the eigenvector $\mathbf{v}_1$ with largest eigenvalue β_1, so that it has the greatest contribution to the sum. By construction, entries of $\mathbf{s}$ take only the values +1 or -1. Thus to attempt to maximise Q under this constraint, we pick values s_i to be of the same sign as the leading eigenvector v_i (so that $v_{1i} > 0$ implies $s_i = +1$ and $v_{1i} < 0$ implies $s_i = -1$). Hence, we have a reasonable approximation to the problem of partitioning a directed network into two communities by maximising the modularity.

Once we have partitioned a network into two communities, we use a 'fine-tuning' step similar to one proposed by Newman [28] as a modification to the Kernighan-Lin algorithm [17]. The fine-tuning switches the community of each node (so $\alpha \to \beta$ or $\beta \to \alpha$) in turn to check whether this increases modularity. If this increases modularity, then the change is kept, otherwise disregarded. We keep looping over the nodes until no more improvement can be made. To keep computation time down, each node is only allowed to change community once. This works well as a fine-tuning step but is not ideal as a method on its own, because it is sensitive to the initial partition [27].

With a method to divide a network into two communities, we then repeat the method on each of the two new communities generated and iterate. After each stage, we need to check whether modularity has increased or not at each division; if not, we do not allow the division to take place. Modularity is calculated on the entire $N \times N$ network, whilst the eigenvalue calculations on $\mathbf{B} + \mathbf{B}^{\mathrm{T}}$ are done on sub-networks formed of members of the community being partitioned. Once modularity can no longer be increased, we have our final communities from the algorithm.

5.3 Extension to Temporal and Multiplex Networks

We can also extend modularity optimisation to networks with multiple layers by utilising the methods suggested by Mucha et al. [24], which we will need for the multiplex and temporal networks. We will use a revision of the method that is appropriate for directed networks. The method works by considering null models for multilayer networks, which are be incorporated into the definition of modularity.

We define the adjacency tensor (A_{ijs}) for the multilayer networks (see Sections 2.5 and 2.6), but now we also must define inter-layer coupling (C_{jrs}) which connects node j in layer r to itself in layer s. For simplicity, we take $C_{jrs} \in \{0, \omega\}$. For our application, we shall always connect nodes to themselves between layers in multiplex networks. In temporal networks, we connect nodes in adjacent layers, so that

$$C_{jrs} = \begin{cases} 1, & \text{if year } r \text{ and year } s \text{ are consecutive,} \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

We define the out-degree in layer s as $k_{is}^{\text{out}} = \sum_{j=1}^{n} A_{ijs}$, in-degree $k_{js}^{\text{in}} = \sum_{i=1}^{n} A_{ijs}$, the total edge weight in layer s as $m_s = \sum_{i,j} A_{ijs}$, the total inter-layer couplings involving s as $c_s = \sum_{i,j} C_{jrs}$, and a normalisation constant $\mu = \sum_s (m_s + c_s)$. If g_{is} is the community of the i^{th} node in the s^{th} layer, then we define modularity for multilayer networks with directed intra-layer edges as

$$Q = \frac{1}{\mu} \sum_{ijrs} \left\{ \left(A_{ijs} - \gamma_s \frac{k_{is}^{\text{out}} k_{js}^{\text{in}}}{m_s} \right) \delta(s, r) + \delta(i, j) C_{jsr} \right\} \delta(g_{is}, g_{js}), \tag{14}$$

where the normalisation constant μ gives us a maximal value of 1 for the modularity and γ_s is used as a resolution parameter, which can be used to indicate relative importance of layers or varied uniformly to enforce larger or smaller communities. To restore the traditional definition of modularity for a single-layer network [30], we set $\gamma_s = 1$ for all s.

As explored by Mucha et al. [24], altering ω has a large impact on the community structure. In the case $\omega = 0$, there are no inter-layer connections between nodes, so communities will be formed only within individual layers. For very large values of ω, edges between layers are strong, but edges between nodes are relatively weak, so each node is in only one community, which defeats the point of considering layered networks compared to aggregating layers. As it is still an open question what choice of ω is best, a range of values will be investigated for multilayer networks.

Now that we have defined modularity for multilayer networks, we need to also extend the Newman-Leicht community-detection method. We do this by turning our tensors $\mathbf{A}$ and $\mathbf{C}$ into one large matrix $\mathbf{B}$ by *tensor flattening* [18]. If we have N nodes in S layers, this yields a $(NS) \times (NS)$ matrix where we consider S lots of $N \times N$ sub-matrices. Diagonal sub-matrices are the adjacency matrices corresponding to that layer (so the s^{th} diagonal sub-matrix has (i, j) entry A_{ijs} for $i, j \in \{1, \ldots, N\}$). For other sub-matrices, if we are in the r^{th} sub-matrix along and s^{th} down, then we have diagonal entries of that sub-matrix given by C_{jrs} for $j \in \{1, \ldots, N\}$ and all other entries are 0.

Note that for multilayer networks, it becomes computationally inefficient to run the Kernighan-Lin 'refinement' step for the community detection algorithm, so this process is no longer run.

6 Comparison of Communities

Once we have found the communities for a network, it may be important to compare them to communities found from other methods or to 'ground truth' partitions. Here we use only a very simple method for comparison between communities, called the Jaccard coefficient [5].

For $v \in V$ let $C_A(v)$ be the community assignment from method A of node v and $C_B(v)$ be the community from method B of node v, then we can define a similarity measure for the partition

$$J = \frac{1}{N} \sum_{v \in V} \frac{|C_A(v) \cap C_B(v)|}{|C_A(v) \cup C_B(v)|}. \tag{15}$$

This gives a value between 0 and 1, with larger values signifying greater similarity.

To give the number some context, it can be compared to a permutation test of the communities. We conduct the test by generating random partitions $\{\tilde{C}_A\}$, $\{\tilde{C}_B\}$ for the null model of random community assignment with the restriction that we keep the same number of nodes in each community as the original partition [40]. This is repeated 10000 times to calculate a z-score for the similarity of our communities in comparison to the null model.

Despite criticism of the Jaccard coefficient due to poor performance for hierarchical partitions [38] (where one partition contains sub-communities of another partition), we use this measure due to it's simplicity and because z-scores provide an easily understood value for how significant the similarity of our partitioning is compared to the null model.

7 Community Structure and Statistics for the National Football League and Major League Baseball

In this section, we consider the communities obtained from community-detection algorithms and the information they give us about mesoscopic structure.

7.1 Single-layer networks

For the single-layer networks, we can firstly examine the distribution of the number of coaches in each community. For coaches in single-layer networks, there is a time element from when they coached not directly represented in the network structure, so that it we investigate whether partitions group coaches from similar eras. We can test this by plotting the *middle year* for each head coach (the middle year of a coaches tenure) within a community in a cumulative diagram.

We also examine how communities have retained edges, which is calculated for some community α by

$$\text{Edges Remaining within community } \alpha = \frac{\text{Edges that start in } \alpha \text{ which remain in } \alpha}{\text{Total outgoing edges from } \alpha}, \tag{16}$$

which is of interest to us to look at the effectiveness of algorithms at capturing a key part of the notion of a community.

Once we have obtained communities, it also of interest to examine homogeneity of various diagnostics between the communities, to look at whether there are inherent differences between communities. The key diagnostics are:

- Total winning fraction for each community, to examine whether a community has more successful coaches than others in absolute winning terms. Note that the NFL and MLB promote parity through measures like a cap on salaries [19], so that a coach should have the opportunity over his tenure to turn a losing team into a winning one. This is unlike the English Premier League where only a handful of teams have the opportunity to win in any given season [36].

- Winning fraction between communities to examine if any community has a particular advantage over others. This could indicate that coaches in a particular community play with tactics which give them an advantage over another community's tactics.

- Coach influence diagnostic, a simple measure to investigate whether communities contain more influential coaches (see Section 4.2).

7.1.1 NFL Binary Network

For the binary network, the community detection algorithm found 7 communities with modularity 0.416. The community sizes are relatively uniform (see Table 3). These communities are also large enough to try and infer some basic facts from them. We also see from Table 3 that all communities recorded between 87% and 95% of outgoing edges remaining within that community.

From Figure 5a, we see that the communities divide into two groups: one containing 4 communities which includes mostly coaches from earlier eras (1960s and 1970s) and another containing 3 communities from later years (1990s and 2000s). Any coaches from the middle years (late 1970s to early 1990s) lie in either group. This illustrates a temporal element to the communities derived from the binary network.

From table 3, we see only small differences in win fraction for communities. The win fraction for matches played between coaches from different communities have also been considered in Table A.10. There are situations where members of one community appear to have advantage over members of another, for example community 2 has a win fraction over 0.545 against 4 of the communities, but below 0.49 for the other 2.

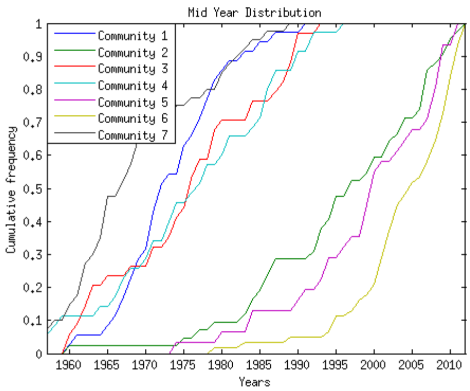
From mean values of the coach influence diagnostic in Table 3, we see there are differences in influence between communities. However this does not correlate with win fraction as the two communities with lowest 'win fraction' have the largest and smallest 'influence'.

7.1.2 NFL Weighted Network

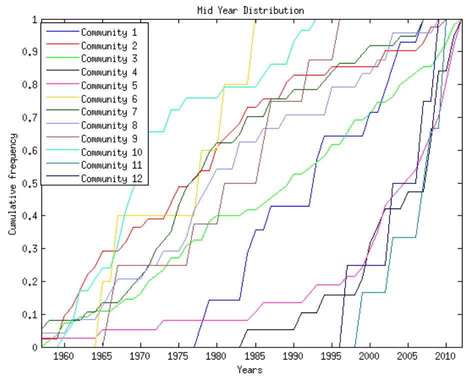
For of the weighted network, the algorithm produced 12 communities with modularity 0.494. We compare the similarity of the communities obtained from the weighted and binary networks with the community comparison score from Section 6. The community comparison score is 0.160, with a z-score with respect

Community Number	Number of Coaches	Edges Held	Coach Influence	Win Fraction
1	35	91.2%	1.07	0.495
2	42	90.0%	1.05	0.505
3	34	90.3%	0.95	0.490
4	35	93.9%	0.96	0.487
5	31	87.1%	1.13	0.486
6	62	94.4%	0.99	0.534
7	40	88.9%	0.86	0.486

Table 3: Communities in the NFL binary network. For an explanation of measures used see Section 7.



(a) NFL Binary Network



(b) NFL Weighted Network

Figure 5: Plotting the 'middle year' of each coaches' tenure.

to the null model of 28.8, which is large enough to indicate a strong statistical significance of the data. It is to be expected that there is a degree of similarity between the two partitions derived given that they have the same network topology.

Unlike the binary case, there are significant differences between the number of coaches in each community (see Table 4). There is also a significant difference in percentage of edge strength that remain within communities, with only 78.6% of all edges remaining in their communities. This could be due to the presence of smaller communities or the partition is not as good as the binary network. We also see from Figure 5b that many communities span more eras than they do in the binary network, for example community 3 spans all eras relatively evenly.

From Table 4, we see a larger difference in results for communities compared to the binary network. Community 6, albeit with only 5 coaches, has a win fraction of only 0.358, which is very poor considering results in the binary network, where no community had below 0.486. Community 10 has more coaches, but still a low win percent at 0.440.

If we look at Table A.11 we see results in matches played between communities, with similar variation

Community Number	Number of Coaches	Edges Kept	Coach Influence	Win Fraction
1	14	57.1%	1.09	0.493
2	41	73.7%	1.12	0.522
3	55	90.0%	1.03	0.508
4	19	77.8%	1.12	0.511
5	37	75.8%	0.86	0.508
6	5	60.0%	0.69	0.358
7	37	77.1%	1.13	0.498
8	24	85.7%	0.88	0.489
9	8	75.0%	0.82	0.477
10	29	88.9%	0.72	0.440
11	6	66.7%	1.54	0.568
12	4	25.0%	1.16	0.521

Table 4: Communities in the NFL weighted network. For an explanation of measures used see Section 7.

in results between different communities. For interactions between communities 4, 7 and 8 we have an interesting interaction where 58% of the time community 4 beats community 7; 56% of the time, 7 beats 8 and 58% of the time, 8 beats 4. This provides a ‘rock-paper-scissors’ type relationship over the 526 matches played between communities, which may be due to tactics employed by the different teams.

We also see large differences in coach influence amongst communities from Table 4. Similar to win fraction, differences are greater than for the binary network. This shows stronger correlations between communities and diagnostics in the weighted network.

7.1.3 MLB Binary Network

For the MLB binary network, the Newman-Leicht method gives 8 communities with modularity 0.486, which is similar to the NFL binary network, with a slightly higher modularity value possibly due to scaling factors from the size of the network. Similar to the NFL binary network a large percentage of edges remain within communities (see Table 5), with all out-edges from community 8 remaining inside. In contrast, we also have a more varied number of coaches per community.

Figure 6a illustrates that there is a temporal element to the partitions like in the NFL binary network. It could also explain community size differences as the largest community contains coaches who started coaching the earliest with a relative sparsity of edges due to small coaching teams. In later years, when coaching teams grew in size (so out-degree increased), communities become smaller in general.

We also see from Table 5 show that there is very little difference between the communities in win fraction, with perhaps even more uniformity than in the NFL binary network. This is also applies to win fraction between communities (see Table A.12); all communities except community 1 have a small amount of variation in results between them.

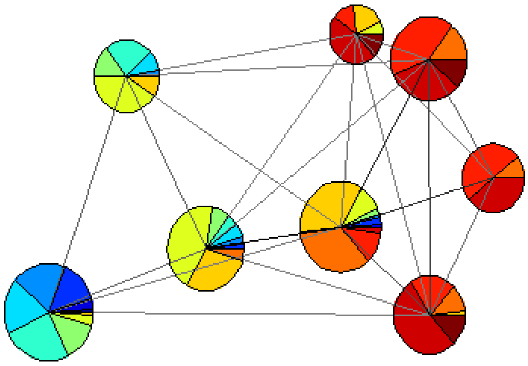
7.1.4 MLB Weighted Network

We found 14 communities in the MLB weighted network with modularity 0.567. Similar to the NFL, the number of coaches in each community is more dispersed in weighted networks than binary networks. There is a similar number of communities compared to the NFL weighted network, but a much higher proportion high proportion (86%) of edges remain within their community. From Figure 6b, we note that the weighted network is similar to the binary network, with communities exhibiting a strong temporal element.

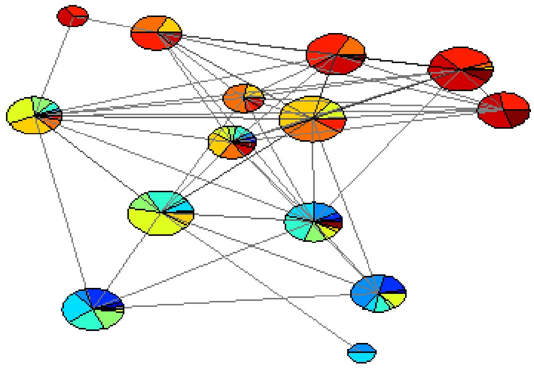
The community comparison score found between the partitions of the MLB weighted and binary networks was 0.180 with a z-score compared to the null model of 55.7. This shows a statistically significant

Community Number	Number of Coaches	Edges Kept	Coach Influence	Win Fraction
1	34	84.3%	0.98	0.508
2	83	92.2%	1.08	0.498
3	63	88.1%	0.89	0.487
4	61	96.4%	1.00	0.498
5	112	94.3%	0.98	0.511
6	47	88.4%	1.02	0.510
7	15	91.8%	1.13	0.515
8	28	100%	1.00	0.485

Table 5: Communities in the MLB binary network. For an explanation of measures used see Section 7.



(a) MLB Binary Network



(b) MLB Weighted Network

Figure 6: Plotting the decade of the 'middle year' of each coaches' tenure in MLB. Communities were placed based on the Fruchterman-Reingold Algorithm [12]. Blue colours represent earlier years and red colours represent later years.

level of similarity with the binary network.

From Table 6 we notice homogeneity between mean win fractions of the baseball managers; in the NFL weighted network, different communities had larger differences in win fraction, but here we see that communities all have a win fraction between 0.46 and 0.52. This should not be surprising as advanced baseball statistics have shown top managers only contribute approximately 5 wins per 168 game season over a 'replacement' manager [8], whilst in the NFL there are often large differences between coaches. When comparing the win fraction between communities (see Table A.13), differences in results are larger than for single communities, but this might be partly due to a smaller sample size of games played.

In Table 6, we give coach influence for the weighted MLB network. Similar to the NFL the smaller communities are at the extremes of influence, but there is still a difference between the larger communities.

Community Number	Number of Coaches	Edges Kept	Coach Influence	Win Fraction
1	36	83.8%	0.86	0.493
2	2	50.0%	0.67	0.485
3	29	84.6%	0.70	0.465
4	48	93.8%	1.00	0.520
5	30	78.6%	1.22	0.508
6	59	90.6%	0.94	0.501
7	18	50.0%	1.07	0.510
8	22	90.5%	1.19	0.506
9	41	84.2%	1.08	0.495
10	56	90.0%	0.94	0.492
11	3	100%	0.83	0.497
12	9	75.0%	1.35	0.510
13	25	87.0%	0.91	0.506
14	65	90.3%	1.08	0.495

Table 6: Communities in the MLB weighted network. For an explanation of measures used see Section 7.

Thus even if some communities do not differentiate themselves in terms of winning, there are differences in their influence.

7.2 Defensive Strategies in the NFL

In the NFL, defences work on one of two base formations, either the 4-3 (4 defensive linemen and 3 linebackers) or the 3-4 (3 defensive linemen and 4 linebackers). This works in a similar way to soccer teams playing formations like 4-4-2 (4 defenders, 4 midfielders and 2 strikers). The key difference in the NFL is defences will only play under one system in any given season, so it is possible to use data on defensive formations for an entire year [1]. This gives a 'ground-truth' community partition for sports coaches that can be imposed.

Some of the 'defensive leaders' have coached more than one method, so we partitioned them into two communities depending on which method they have coached more (most coaches have only coached one method). This gives a network partition of 148 coaches coaching primarily 4-3 and 62 coaching primarily 3-4.

We found that the partition by defensive strategy gives a modularity of 0.127. We then compare it to the result found from running the Newman-Leicht method with Kernighan-Lin refinement, but with the restriction of only running the algorithm once to give two communities. We only consider the two communities as our data only splits into two communities. This gave a modularity of 0.334, so that the ratio of 'ground-truth' data to the algorithm is $\frac{0.127}{0.334} = 0.380$.

We can further restrict the modularity optimisation method by forcing the same split of coaches for each community, to give a community of 148 coaches and one of 62. We add a step after Kernighan-Lin refinement so that each member of one community is placed into the other in turn to find out the modularity change for a single node. If we need to move k coaches from this community in order to get required community sizes then we move the k coaches that cause the smallest drop in modularity. This gives two communities of correct size. As either community could be chosen to gain/lose nodes, this process is repeated but with role of gaining/losing switched to give two community partitions of correct sizes.

Under this restriction, modularity from the community detection algorithm is now either 0.270 or 0.298, depending on which community gains/loses nodes, giving a ratio of 'ground truth' data to optimised data of 0.470 and 0.426 respectively. This ratio score is a high one for the partition into defensive categories. This illustrates that partitioning coaches in our 'defensive' network based upon defensive formation is a reasonable way of partitioning the network of defensive coaches into communities.

We can also apply the 'community comparison' methods from Section 6 to examine how similar the communities are when derived from 'ground-truth' data compared to algorithmic methods. Running the algorithm given above, this gives comparison scores of 0.408 and 0.386. In comparison, 10000 Monte-Carlo simulations on communities of the same size gave a mean score of 0.397. This gives z-scores of 1.26 and -1.21, neither of which show a statistically significant amount of similarity between the partitions.

7.3 Multilayer Networks

In the multilayer networks, there exists freedom in the choice of inter-layer coupling strength ω (see Section 5.3), which will be investigated. The criteria on which we will investigate are:

- The number of communities.

- Modularity- note these values will be higher than single-layer networks due to scaling factors (see Section 5.1).
- The mean number of layers that a community spans. We say that a layer is in the *span* of a community if the community is represented by at least one coach in that layer of the network.
- The mean number of coaches that a community spans. Similarly, a coach is in the *span* of a community if the coach is a member of that community in at least one layer.

As it is still an open question for choice of ω and it is not practical to study a large range of values, we will choose ω to try to find a mix of *inter-layer communities* (communities that span multiple layers) and *intra-layer communities* (communities which span many coaches). ω will be found by picking a range of values and picking ω which maximises the geometric mean of L (mean number of layers a community spans) and C (mean number of coaches a community spans). This is still an arbitrary choice but it is preferable to picking ω at random as it allows us to investigate communities at the balance between inter- and intra-layer communities and so we do not pick ω towards extremal values (see Section 5.3).

The partition of communities given by this value of ω will be examined under applicable diagnostics given for the single-layer networks (see Section 7.1). As we have relaxed the restriction of coaches belonging to multiple communities, to calculate the mean we consider the weighted average from the number of times a coach appears in each community. For example, if coach A appears in a community twice (in two different layers) and coach B appears once, then coach A has twice the weighting for his scores than coach B.

7.3.1 NFL Temporal Network

From Table 7, we see that as we increase the value of ω the number of coaches goes down but also there is no noticeable increase in layers spanned per community. This could possibly be explained by the decrease in number of coaches per community reducing the potential to reach extra layers. We choose $\omega = 2$ to investigate further as amongst values studied it maximised geometric mean. The diagnostics for $\omega = 2$ are given in Table A.14.

For $\omega = 2$, the number of coaches represented in each community is quite varied, with values ranging from 6 to 123, which is a similar distribution of community sizes to the weighted network. The larger

ω	Communities	Avg Layer	Avg Coach	Modularity	Geometric Mean
0.05	13	12.3	42.5	0.275	22.9
0.1	20	13.8	39.5	0.431	23.3
0.2	58	13.3	21.8	0.701	17.0
0.5	30	16.5	31.3	0.796	22.7
1	51	13.9	17.3	0.768	15.5
2	27	21.9	28.3	0.843	24.8
5	65	15.4	11.1	0.871	13.1
10	89	17.2	7.7	0.910	11.5
20	121	16.9	5.36	0.929	9.5
50	106	14.8	4.67	0.755	8.3

Table 7: Investigating different choices of ω for the NFL temporal network. Avg Layer is mean layers spanned per community and Avg Coach is mean coaches spanned per community.

ω	Communities	Avg Layer	Avg Coach	Modularity	Geometric Mean
0.01	5	1.4	202	0.578	16.8
0.05	5	1.6	199	0.579	17.8
0.1	5	1.80	196	0.572	18.8
0.2	6	1.83	180	0.588	18.1
0.5	6	1.83	174	0.560	17.9
1	5	1.80	182	0.522	18.1
2	4	1.75	186	0.531	18.0
5	6	2	142.6	0.585	16.9
10	21	2	34.5	0.669	8.3

Table 8: Investigating different choices of ω for the MLB multiplex network. Avg Layer is mean layers spanned per community and Avg Coach is mean coaches spanned per community.

variety in values for both the mean coach influence and win fraction are also similar to the weighted network. They both show a strong correlation between community membership and the 'success' of coaches and the teams they coach.

7.3.2 MLB Multiplex Network

We now consider the multiplex network where the layers are given by coach→player tutelage and coach→coach tutelage. In contrast to the NFL temporal network, as ω increases both layers spanned increases and the number of coaches spanned decreases. This could be explained by the difference between 2 layers in the multiplex network compared to 56 for the temporal network. To minimise geometric mean, $\omega = 0.1$ is chosen (see Table 8).

Similar to the binary networks, community size are all roughly of the same order. We find from that there are only very small differences in win fraction between the communities (see Table A.15), which is very similar to other MLB networks.

To calculate coach influence diagnostic for the multiplex network, we consider out-degree as the aggregation of both layers in the network. There are variations in the measure, similar to the binary network, with community 3 made up only of coaches from the player→coach layer performing the worst.

7.3.3 MLB Temporal Network

The MLB temporal network for all 1907-2012 coaches proved to be too computationally heavy, so we only considered the past 50 years (1962-2012) of data to be considered for analysis.

From Table 9, the MLB temporal network produces similar community structure to the NFL temporal network as ω is increased, with layers spanned per community varying up and down in addition to coaches spanned decreasing. Out of values for ω investigated, $\omega = 0.5$ would be the best choice to maximise the geometric mean. More detailed results for $\omega = 0.5$ are given in Table A.16.

A more thorough investigation of $\omega = 0.5$ gives results similar to the communities derived for the MLB weighted network. The differences in win fraction between communities is neglible, and there is a reasonable amount of difference in coach influence (for example community 14 spans 99 coaches yet have a coach influence score of only 0.78). We also note a large difference in the number of coaches spanned by each community.

ω	Communities	Avg Layer	Avg Coach	Modularity	Geometric Mean
0.1	11	9.1	44.9	0.431	22.0
0.2	34	12.2	26.9	0.539	18.1
0.5	19	19.6	43.9	0.670	29.3
1	49	13.9	17.6	0.853	15.6
2	51	13.9	15.4	0.859	14.6
5	47	18.0	16.3	0.88	17.1
10	83	14.6	9.1	0.914	11.5
20	80	16.8	9.0	0.921	12.3
50	135	13.6	5.1	0.937	8.3

Table 9: Investigating different choices of ω for the MLB temporal network. Avg Layer is mean layers spanned per community and Avg Coach is mean coaches spanned per community.

8 Summary

In this dissertation, we used data sets from the NFL and MLB to examine networks of coaching lineages. We examined whether a coaches' lineage influences whether their teams are successful. We found that in MLB, amongst the networks studied, there is little difference in win fractions between communities. This was not considered surprising given that advanced baseball statistics agreed that managers do not have a large influence on the results of matches [8]. In contrast, for the NFL weighted and temporal networks, there were more significant differences in win fraction. This suggests that the coaches that an assistant works under affects future success. This could be as a result of different tactics being handed down from coach to coach.

Looking at results between communities, there were reasonable differences in winning percentage in the NFL, which may be an example of some coaches having an advantage over some communities but not over others.

By studying the mean of the 'coach influence' diagnostic for a community we illustrated that there can be some differences between communities for all networks studied, especially in the weighted networks. This might imply that some communities are more important in how tactics developed in the sports than others if their coaches were more influential.

The network properties of communities generated from the algorithm were also studied. In both the NFL and MLB, there was a statistically strong correlation between communities generated in the binary and weighted networks. In general, binary networks produced fewer communities with a smaller range in community size compared to the weighted networks. Also, communities from the binary networks retained edges within communities better than weighted networks. This could show that the communities generated in the binary network are more sensible when considering the definition given in Section 2.8.

When considering the network of 'defensive leaders' in the NFL with known partitions for formations, we showed that communities derived from this information produce a small but not insignificant modularity score when compared to the score that can be made from community-detection algorithms. This suggests the partition formed from different strategies meet the definition given in Section 2.8. However, the community detection algorithms did not give us communities which have a statistically significant degree of similarity to the communities derived from ground-truth data.

Multilayer networks were also studied in this dissertation, with a range of values for inter-slice coupling ω considered. As this value increased in temporal networks, the number of layers spanned per community varied up and down, unlike the multiplex baseball network, where it increased. In both multiplex and temporal networks, the number of coaches spanned descreased as ω increased. Where a single value for ω was studied more closely, in general the multiplex network behaved like the binary single-layer networks and the temporal networks behaved like the weighted single-layer networks.

9 Potential Extensions

There are numerous ways that we could build on our work in this dissertation. These can be split into two areas: additional data and different methods of analysing the data.

From the perspective of additional data, it would be of interest to examine a greater variety of sports. For example, many soccer managers learn how to manage by playing under other managers rather than as an assistant, so this could be studied as a multiplex network as we did for MLB managers. For more data in American football, the database could be expanded to include American college football in addition to the professional game, as hundreds of colleges play football [25], most with large coaching staffs, so theoretically networks of thousands of football coaches could be studied.

We could also do additional research with different methods of analysis. One of the simplest ideas would be to study different types of community-detection algorithms; although methods that we used were useful, there are many different methods to detect communities for directed networks [21]. We could also study whether successful coaches are more likely to be connected to other successful coaches by considering assortative mixing between coaches [11]. There are also opportunities to apply statistical methods by considering statistical error of results or considering permutations in the original networks to test for robustness of communities discovered.

One could also consider multiplex coaching networks made up different types of assistant coaches that can exist in both MLB and the NFL, with each layer representing the different types of assistant. For example, in the MLB, it is often remarked that bench coaches are seen as future managerial material or former managers working as mentors [23]. These links could be to examine whether they bring an increased likelihood of success for coaches involved.

10 Acknowledgements

I would like to extend many thanks and much gratitude to my supervisor for the many constructive comments throughout the process of this dissertation and also for agreeing to be my supervisor. Comments and feedback from the presentation I gave to his group were very helpful.

I would also like to thank the data sources used for this: retrosheet (`http://www.retrosheet.org`) for all MLB data; pro football archives (`http://www.profootballarchives.com`) for all NFL coaching and results data and pro football reference (`http://www.pro-football-reference.com`) for NFL formations data.

Final thanks to NetWiki (`netwiki.amath.unc.edu`) for providing code for the visualisation of graphs.

References

[1] Pro Football Archives. Web page at http://www.profootballarchives.com. Accessed: 10/08/2013.

[2] Pro-Football-Reference. Web page at http://www.pro-football-reference.com. Accessed 10/11/2013.

[3] Retrosheet. Web page at http://www.retrosheet.org. Accessed: 10/08/2013.

[4] A. Arenas, J. Duch, A. Fernández, and S. Gómez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9:176, 2007.

[5] A. Ben-Hur, A. Elisseeff, and I. Guyon. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing*, 7:6–17, 2002.

[6] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[7] T. Callaghan, P. J. Mucha, and M. A. Porter. Random walker ranking for NCAA Division I-A football. *American Mathematical Monthly*, 114(9):761–777, 2007.

[8] A. Darowski. Manager wins above expectancy. Web page at http://darowski.com/hall-of-wwar/expectancy/, 2012.

[9] A. S. Fast and D. D. Jensen. The NFL coaching network: Analysis of the social network among professional football coaches. *Presented at AAAI Fall Symposia on Capturing and Using Patterns for Evidence Detection*, 2006.

[10] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75–174, 2010.

[11] J. G. Foster, D. V. Foster, P. Grassberger, and M. Paczuski. Edge direction and the structure of networks. *Proceedings of the National Academy of Sciences*, 107:10815, 2010.

[12] T. M. J. Fruchtermann and E. M Reingold. Graph drawing by force-directed placement. *Software, Practice and Experience*, 21:1129–1164, 1991.

[13] B.H. Good, Y.A. De Montjoye, and A. Clauset. The performance of modularity maximization in practical contexts. *Physical Review E*, 81:046106, 2010.

[14] P. Holme and J. Saramäki. Temporal networks. *Physics Reports*, 519(3):97–125, 2012.

[15] S. Kacsmar. Dissecting the coaching trees in the NFL today. Web page at http://bleacherreport.com/articles/1663695-dissecting-the-coaching-trees-in-the-nfl-today, 2013.

[16] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information Processing Letters*, 31:7–15, 1988.

[17] B.W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970.

[18] M. Kivelä, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. *Preprint arXiv:1309.7233*, 2014.

[19] M. Lawrence. Football 101: The salary cap. Web page at `http://football.calsci.com/SalaryCap.html`, 2005.

[20] E. A. Leicht and M. E. J. Newman. Community structure in directed networks. *Physical Review Letters*, 100:118703, 2008.

[21] F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 533(4):95–142, 2013.

[22] R.D. Malmgren, J. M. Ottino, and L. A. N. Amaral. The role of mentorship in protégé performance. *Nature*, 463:622–626, 2010.

[23] S. Miller. Next to the manager, but a bit ahead. Web page at `http://www.nytimes.com/2012/08/19/sports/baseball/in-major-league-baseball-bench-coaches-assume-growing-role.html`, 2012.

[24] P. J. Mucha, T. Richardson, K. Macon, M. A Porter, and J.-P. Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *Science*, 328(5980):876–878, 2010.

[25] NCAA. NCAA football standings. Web page at `http://www.ncaa.com/standings/football/fbs`. Accessed 17/03/2014.

[26] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[27] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[28] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences USA*, 103:8577–8582, 2006.

[29] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[30] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[31] G. Palla, A.-L. Barabási, and T. Vicsek. Quantifying social group evolution. *Nature*, 446:664–667, 2007.

[32] M. A. Porter, P. J. Mucha, M. E. J. Newman, and A. J. Friend. Community structure in the United States House of Representatives. *Physica A*, 386(1):414–438, 2007.

[33] M. A. Porter, J.-P. Onnela, and P. J. Mucha. Communities in networks. *Notices of the American Mathematical Society*, 56(9):1082–1097, 1164–1166, 2009.

[34] Pro Football Hall of Fame. Don Shula. Web page at `http://www.profootballhof.com/hof/member.aspx?PLAYER_ID=194`. Accessed 22/03/2014.

[35] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74:016110, 2006.

[36] D. Rockerbie. Exploring inter-league parity in North America: the NBA anomaly. *MPRA Paper*, 2012.

[37] S. Saavedra, S. Powers, T. McCotter, M. A. Porter, and P. J. Mucha. Mutually-antagonistic interactions in baseball networks. *Physica A*, 389(5):1131–1141, 2010.

[38] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1:27–64, 2007.

[39] Sports Illustrated. When NFL coaches were players. Web page at `http://sportsillustrated.cnn.com/nfl/photos/1301/when-nfl-coaches-were-players/`, 2013.

[40] A. L. Traud, P. J. Mucha, and M. A. Porter. Social structure of Facebook networks. *Physica A*, 391(16):4165–4180, 2012.

[41] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

A Supplementary Tables

A.1 Results Comparisons

Tables for the results comparison between communities. The entry in (i, j) represents the wins fraction of coaches in community i in matches against community j. Where fewer than 100 games have been played between communities or for matches between communities no result is given.

	1	**2**	**3**	**4**	**5**	**6**	**7**
1	-	0.455	0.534	0.459	0.505	0.518	0.503
2	0.545	-	0.566	0.560	0.489	0.469	0.549
3	0.466	0.434	-	0.520	0.523	0.445	0.511
4	0.541	0.44	0.480	-	0.471	0.428	0.518
5	0.495	0.511	0.477	0.529	-	0.466	-
6	0.482	0.531	0.550	0.572	0.534	-	0.539
7	0.497	0.451	0.489	0.482	-	0.461	-

Table A.10: Results comparison between communities in the NFL binary network. See above for an explanation.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	-	0.517	0.465	0.526	0.404	-	0.540	0.512	-	-	-	-
2	0.483	-	0.522	0.519	0.482	-	0.526	0.521	0.457	0.575	-	-
3	0.535	0.478	-	0.463	0.502	-	0.515	0.513	0.529	0.592	0.469	-
4	0.474	0.481	0.537	-	0.535	-	0.584	0.420	-	-	0.450	-
5	0.596	0.518	0.498	0.465	-	-	0.486	0.491	-	0.635	0.421	-
6	-	-	-	-	-	-	-	-	-	-	-	-
7	0.461	0.474	0.485	0.416	0.514	-	-	0.559	-	0.543	-	-
8	0.488	0.479	0.487	0.580	0.509	-	0.441	-	-	0.506	-	-
9	-	0.543	0.471	-	-	-	-	-	-	-	-	-
10	-	0.425	0.408	-	0.365	-	0.457	0.494	-	-	-	-
11	-	-	0.531	0.550	0.579	-	-	-	-	-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-

Table A.11: Results comparison between communities in the NFL weighted network. See above for an explanation.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	-	0.497	0.527	0.426	0.503	0.408	0.508	-
2	0.503	-	0.513	0.508	0.458	0.491	0.492	0.488
3	0.473	0.487	-	0.485	0.491	0.508	0.476	0.507
4	0.574	0.492	0.515	-	0.506	0.493	0.486	0.513
5	0.497	0.542	0.509	0.493	-	0.498	0.522	0.514
6	0.592	0.509	0.492	0.507	0.502	-	0.490	0.529
7	0.492	0.508	0.524	0.514	0.478	0.510	-	0.559
8	-	0.512	0.493	0.487	0.486	0.471	0.441	-

Table A.12: Results comparison between communities in the MLB binary network. See above for an explanation.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	-	-	0.519	0.493	0.492	0.467	0.499	0.603	0.487	-	-	-	-	0.533
2	-	-	0.487	0.433	-	0.437	-	-	-	-	-	-	-	-
3	0.481	0.513	-	0.445	0.486	0.468	0.514	-	-	-	-	-	-	0.502
4	0.507	0.567	0.555	-	0.511	0.502	0.534	-	0.533	-	-	-	0.459	0.520
5	0.508	-	0.514	0.489	-	0.508	0.505	0.479	0.537	0.529	-	0.460	0.493	0.511
6	0.533	0.563	0.532	0.498	0.492	-	0.509	0.429	0.522	0.468	-	0.462	0.535	0.467
7	0.501	-	0.486	0.466	0.495	0.491	-	0.500	0.502	0.529	0.414	0.526	0.518	0.529
8	0.397	-	-	-	0.521	0.571	0.500	-	0.505	0.505	0.525	0.483	0.499	0.509
9	0.513	-	-	0.467	0.463	0.478	0.498	0.495	-	0.487	0.523	0.488	0.489	0.526
10	-	-	-	-	0.471	0.531	0.471	0.495	0.513	-	0.508	0.473	0.484	0.483
11	-	-	-	-	-	-	0.586	0.475	0.477	0.492	-	0.592	0.504	0.532
12	-	-	-	-	0.540	0.538	0.474	0.517	0.51	0.527	0.408	-	0.544	0.481
13	-	-	-	0.541	0.507	0.465	0.482	0.501	0.511	0.516	0.496	0.456	-	0.505
14	0.467	-	0.498	0.480	0.489	0.533	0.471	0.491	0.474	0.517	0.468	0.519	0.495	-

Table A.13: Results comparison between communities in the MLB weighted network. See above for an explanation.

A.2 Tables of Communities in Multilayer Networks

Community Number	Coaches	Layers	Coach Influence	Win Fraction
1	37	21	1.00	0.526
2	42	27	1.29	0.518
3	38	30	1.16	0.514
4	68	24	1.39	0.543
5	39	26	0.89	0.505
6	34	29	0.82	0.491
7	23	16	0.79	0.496
8	9	16	0.86	0.490
9	13	12	0.94	0.490
10	24	18	0.80	0.485
11	19	27	0.88	0.511
12	10	16	1.30	0.557
13	23	20	1.06	0.532
14	34	36	1.27	0.512
15	28	23	0.85	0.521
16	37	21	0.80	0.520
17	17	20	0.68	0.448
18	24	17	0.83	0.497
19	45	15	0.83	0.510
20	6	14	1.03	0.539
21	8	17	1.48	0.526
22	11	21	1.71	0.562
23	9	10	0.71	0.395
24	21	20	1.35	0.539
25	123	55	0.73	0.476
26	11	17	0.90	0.510
27	10	22	0.66	0.509

Table A.14: Communities in the NFL temporal network for $\omega = 2$. ‘Coaches’ gives the number of coaches which are a member of that community in one or more layers. ‘Layers’ refers to the number of layers which a community has at least one member in.

Community Number	Coaches	Layers	Coach Influence	Win Fraction
1	156	2	0.94	0.504
2	263	2	0.94	0.501
3	101	1	0.77	0.485
4	283	2	1.06	0.500
5	178	2	1.21	0.508

Table A.15: Communities for $\omega = 2$ in the MLB multiplex network. 'Coaches' gives the number of coaches which are a member of that community in one or more layers.

Community Number	Coaches	Layers	Coach Influence	Win Fraction
1	23	8	1.20	0.508
2	33	7	0.83	0.494
3	13	9	0.78	0.509
4	30	13	0.86	0.499
5	32	11	1.31	0.520
6	46	10	0.99	0.500
7	39	18	1.23	0.510
8	29	17	1.11	0.516
9	52	32	1.09	0.513
10	34	16	0.97	0.498
11	24	15	0.79	0.478
12	17	13	0.80	0.476
13	11	11	0.65	0.488
14	99	49	0.78	0.502
15	9	14	0.85	0.523
16	34	23	1.34	0.515
17	37	27	0.67	0.484
18	42	29	1.10	0.503
19	228	50	1.00	0.505

Table A.16: Communities in the NFL temporal network for $\omega = 0.5$. 'Coaches' gives the number of coaches which are a member of that community in one or more layers. 'Layers' refers to the number of layers which a community has at least one member in.

B Data Cleaning

The cleaning of data was done entirely by myself on Python exporting data from the internet, which was then converted into a .mat file to be used by MATLAB.

To find the coaching networks for the NFL and MLB, the following process was followed for both retrosheet and profootballarchives:

1. Find a list of all coaches and download names with corresponding URLs.

2. For each coach download all information on their page.

3. Clean this data to give a list of triples (team,year,role) for every year in which a coach coached.
4. Disregard any coaches that were never head coach.
5. Link head coaches to any other coaches that coached for the same team in the same year.
6. Export the adjacency matrix where a 1 in (i, j) represents coach j working for coach i.

To add defensive formations data, I did the following:

1. Find out who the leader of the defence was in each year (either head coach or defensive coordinator) and form the matrix as above.
2. Find the defensive formation used in each year by each team and associate it with the 'defensive leader'.
3. If different formations were used by the same coach associate the one which was used more.
4. Export a vector with the partition based on defensive formation where each entry is a single coaches' associated formation.

To gain results data, a different process was followed for retrosheet and pro-football-reference:

1. Find the result of all matches that have ever been played.
2. Associate the result of every match with the head coach of both teams involved.
3. Be careful when dealing with teams which had multiple head coaches in the same year.
4. Export a matrix where each row/column represents a coach and the entry in (i, j) is the number of times coach i won against coach j.

C Sample Code for Modularity and Newman-Leicht Method

All of the coding necessary for this dissertation was done by me, except for any of the graphics packages. Here I present sample code for the backbone of this dissertation: modularity and the Newman-Leicht method, both of which were coded in MATLAB.

C.1 Modularity

```
function [Q,B] = modularity(A,c)
% The modularity function returns the modularity of an adjacency matrix A
% which is split into communities with the vector c
% OUTPUT modularity Q and modularity matrix B

% We define m as the total number of edges in a network
m=sum(sum(A));
len=size(A,1);

% Vectors of in-degree and out-degree
in_vect=sum(A,2);
out_vect=sum(A,1);
% Get an 'in_out' matrix, essentially weights for the configuration model for finding B
```

```
in_out=in_vect*out_vect;
% Set B is the modularity matrix
B=A-(in_out/m);

% get the 'delta matrix' which returns a 1 if i and j share communities and
% 0 otherwise

delta_mat=zeros(len);
for i=1:max(c)
    k=c==i;
    delta_mat(k,k)=1;
end

% Form the 'martix' of Q, so that Q is just the sum over this matrix
Q_mat=(1/m)*(B.*delta_mat);
Q=sum(sum(Q_mat));
end
```

C.2 Newman-Leicht Method

The coding for the Newman-Leicht Method includes the 'split' function which splits a community into two, and the 'adjusted_kernighan_lin' (not shown; this is only intended to be a presentation of some sample code).

```
function [c] = eig_mod_refine(A)
% Input the adjacency matrix A which we partition into communities
% using the Newman-Leicht method with adjusted Kernighan-Lin.
% The vector c gives the community number for each inputted coach

n_coaches=size(A,1);

% c is the final community vector, n_comms gives the total number of
% communities as the algorithm runs and comm_considered is the community
% currently being split
c=ones(n_coaches,1);
n_comms=1;
comm_considered=1;

% Find the modularity
Q_last=modularity(A,c);

% The main while loop, loop until all communities have had a split attempted
while comm_considered<=n_comms
    % Find all coaches in the community considered and form a submatrix
    % of them to run the split algorithm on
    k=find(c==comm_considered);
    Asub=A(k,k);
    csub=split(Asub);  %Gives 2 community assignments '1' or '2'
    csubext=zeros(n_coaches,1);
    csubext(k)=csub;
    % Form a new 'c' vector
    cnew=c;
    i=csubext==2;
    cnew(i)=n_comms+1;
    cnew=adjusted_kernighan_lin(A,c==comm_considered,cnew); %Optimises the split
```

```
    % Now we test to see if the new communities increase the modularity
    % score
    Qnew=modularity(A,cnew);
    % If the new modularity is higher than old set results as best current estimate
    if Qnew>Q_last
        Q_last=Qnew
        n_comms=n_comms+1;
        c=cnew;
    else comm_considered=comm_considered+1;
    end
end
end

%-------------------------------------------------------------------------

function [c] = split(A)
% The split function partitions an adjacency matrix A into 2 groups given by a
% vector c which is 1 for one community and 2 for the other

% Total number of edges
m=sum(sum(A));

% Find B in exactly the same way as the 'modularity' calculation
in_vect=sum(A,2);
out_vect=sum(A,1);
in_out=in_vect*out_vect;
B=A-in_out/m;

% Now we want to find the eigenvector of B+B' associated with the largest
% positive eigenvalue, denoted b
[b,~]=eigs(B+B');
b=b(:,1);

% c is our vector for showing where the split occurs
c=zeros(size(b,1),1);
for i=1:size(b)
    if b(i)>0
        c(i)=1;
    % If eigenvector has zero entries, sign of disconnectedness so will want
    % to reconsider at next 'split' attempt
    elseif b(i)<=0
        c(i)=2;
    end
end
end
```