Problem Chosen	2020 MCM/ICM Summary Sheet	Team Control Number
D		2013984

Soccer Originated Teamwork Model

In interconnected contemporary societies, effective teamwork is becoming more and more essential due the the increasingly complex tasks. Teams struggling with the most challenging problems are usually made up of people with diverse expertise and varied perspectives.

Competitive team sports are one of the most informative settings to explore team processes. Soccer, one of the most popular team sports, has aroused tremendous research interest. Compared with previous works in this field, our team employ a more microcosmic insight which is a necessity for a better understanding of the interactions between players. Based on the data of the Huskies, we create the **Whole-match Network** to have a general idea of the macroscopic match process. We also cut the passings into topological sequences based on the shift of the possession of ball. Each sequence here represents a continuous attack or defense process. Analysis on the separated sequences allows for a deeper understanding of how the ultimate situation was gradually formed. Based on the sequences, our team propose the **Sequence Network**. Motifs in Sequence Network allows for a better knowledge of the dyadic and triadic configurations.

Based on our model, we provide indicators for individual performance from a diversity of perspectives. We use **closeness centrality** to measure a player's connect with teammates, **betweenness centrality** to measure the significance of a player in the connection of other teammates, and **pagerank** to measure the 'popularity', or trust from teammates, of a player. As for team performance, we propose a **motif contribution score** learned from the data of the Huskies. We also use **clustering coefficients** to measure the extent to which the members cluster. Based on these indicators, we provide suggestions for the couch with the hope that they can perform better in future matches.

Furthermore, we expand the passing network in the soccer field to the communicating network in the generalized team. Specifically, we define the number of e-mails that are sent during a certain period as the weight of the communicating network. To achieve a better team performance, various strategies such as paying more attention to the effective motifs are proposed.

Keywords network theory, motif, team cooperation

Contents

1 Introduction

In interconnected contemporary societies, challenges are becoming more and more complex. Teams struggling with the most challenging problems are usually made up of people with diverse expertise and varied perspectives. Interdisciplinary teams are able to perform complex tasks with individual efforts as well as a sequence of contributions of teammates.

Competitive team sports are one of the most informative settings to explore team processes. Soccer, one of the most popular team sports, has aroused tremendous research interest. Previous research typically focus on individual players, especially those with strong personal ability[1]. Team cooperation, including the micro or macro patterns, deserves more attention[2][3].

There have been attempts to quantitatively evaluate the interactions. Peña et al. generate passing networks from the data of a whole match[4]. Gürsakal et al. analyze the characteristics of the subgraphs and find some motifs are extremely helpful against particular rivals[5]. Another perspective based on such network is to focus on the macro strategies when drafting the tactics before the match[6].

A vast majority of researches analyse team work simply according to the ultimate result of a game, which is too static for a dynamic sport like soccer. A more microcosmic insight is necessary to achieve a better understanding of the interactions between players. Based on the data of the Huskies, we create the **Whole-match Network** to have a general idea of the overall performance. We also cut the passings into topological sequences based on the shift of the possession of ball. Analysis on the separated sequences allows for a deeper understanding of how the ultimate situation was gradually formed. Based on the sequences, our team propose the **Sequence Network**. Motifs in Sequence Network allows for a better knowledge of the dyadic and triadic configurations.

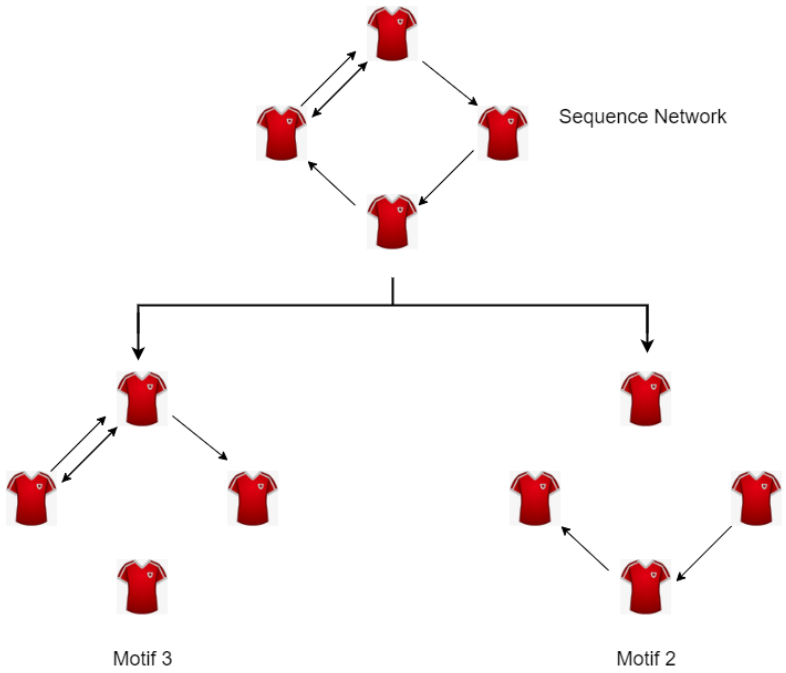

Figure 1: A simplified visualization for finding motifs

Based on our model, we provide indicators for individual performance from a diversity of perspectives. We use closeness centrality to measure a player's connect with teammates, betweenness

centrality to measure the significance of a player in the connection of other teammates, and pagerank to measure the 'popularity', or trust from teammates, of a player. As for team performance, we propose a motif contribution score learned from the data of the Huskies. We also use clustering coefficients to measure the extent to which the members cluster. Based on these indicators, we provide suggestions for the couch with the hope that they can perform better in future matches.

By far, we have been able to consider group dynamics in soccer matches. Similarly, researches in the team networks of the business reveals that effective interaction between team members could also make contributions to the whole team's performance.[7] We expand the passing network in the soccer field to the communicating network in the generalized team. Specifically, we define the number of e-mails that are sent during a certain period as the weight of the communicating network. To achieve a better performance of the team, various strategies such as paying more attention to the effective motifs are proposed.

2 Ball Passing Model

Soccer players are connected to each other with ball passing. The very basic idea is to illustrate a match where each node stands for an individual player, and the weight of edges are decided by the number of passings between the end nodes. We are interested in the overall performance during the whole match. We are also interested in the patterns which happens within small time periods.

Therefore, our model can be divided into the following two parts. The first part is the **Whole-match Network**, which allows for the match-level evaluation. From this network we hope to learn the overall performance of the team as well as each individual player. The second part is the **Sequence Network**. In this part we divide the whole match into different sequences based on shifts of the possession of the ball. We then study the Sequence Network to learn more about the cooperation patterns.

2.1 Whole-match Network

Based on the passings in a match, an intuitive way to construct a network is to use nodes to represent players and the edges to represent the passings between each pair of teammates. Take Match 1 as an example. With the width of each line representing the weight, Fig.2 gives a general idea of the team. It seems D_1 and D_2 are good partners while D_3 and D_4 would not cooperate so much.

This network, however, neglects the direction of each passing. Our Whole-match Network is based on this architecture, but the edges are directed. Denote the number of players as k and we have the following definitions.

Definition 1. *The **Whole-match Network** $Net_W = (P, E)$ is a directed graph where P is the set of **nodes** and E is the set of **edges**. $P = \{p_1, p_2, \cdots, p_k\}$ represent the k **players** in the team. $E = \{\cdots, (p_i, p_j, w_{ij}), \cdots\}$ is the set of edges and w_{ij} is the number of **passings** from player p_i to player p_j.*

Definition 2. *The **adjacency matrix** $A = (a_{ij})^{k\times k}$ represent the number of passings between players.*[1] *For any $1 \le i, j \le k$, $a_{ij} = w_{ij}$.*

[1]Considering the directions, the adjacency matrix A is not necessarily symmetric.

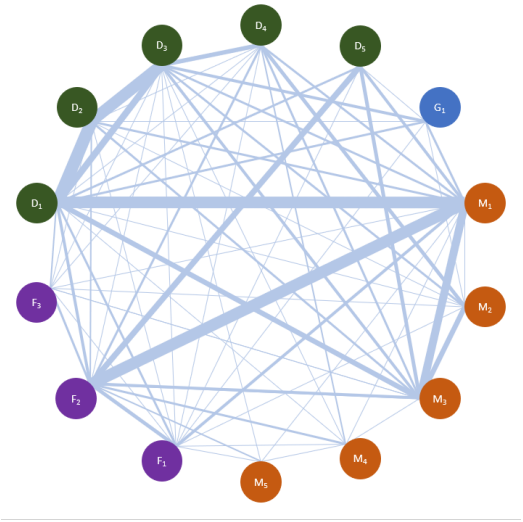

Figure 2: Undirected graph for the Huskies in Match 1

Definition 3. *The matrix* $E = (\epsilon_{ij})^{k\times k}$ *is the none-weighted adjacency matrix where*

$$\epsilon_{ij} = \begin{cases} 0 & , a_{ij} \neq 0 \\ 1 & , a_{ij} = 0 \end{cases}.$$

The number of passings provides much information for the cooperation between players. More passings from player p_i to player p_j means player i and j have better cooperation. Thus we define a matrix L to indicate the 'distance' on the Whole-match Network.

Definition 4. *The distance matrix* $L = (l_{ij})^{k\times k}$ *on the Whole-match Network is a measurement of nodes in the network where*

$$l_{ij} = \begin{cases} 0 & , i = j \\ \frac{1}{a_{ij}} & , i \neq j \end{cases}.$$

Some players may not be directly connected in the network. But they can be connected with some intermediate player. Some far-away players might be more closely connected through some other teammates. Thus we define the geodesic matrix D to demonstrate how players are connected with the assist of other teammates.

Definition 5. *The geodesic matrix* $D = (d_{ij})^{k\times k}$ *on the Whole-match Network is the smallest distances between nodes where*

$$d_{ij} = \min_{k_1,k_2,\cdots,k_m} l_{ik_1} + l_{k_1k_2} + \cdots + l_{k_m}l_j.$$

Fig.3 is a visualization of our Whole-match Network. The size of the node is decided by closeness centrality[2] of the player.

[2]Closeness centrality is an indicator for individual performance and will be discussed in the following section.

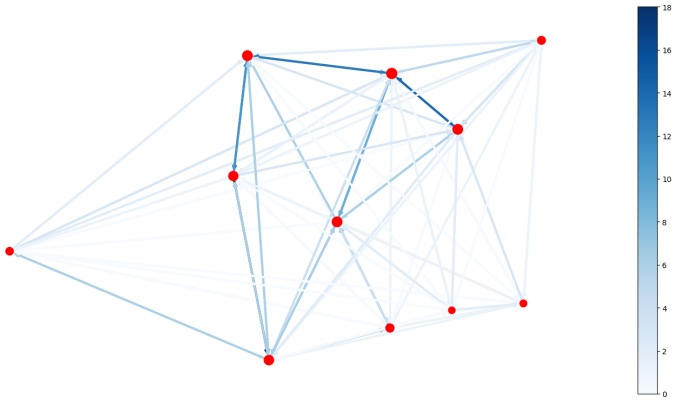

Figure 3: Whole-match Network of the Huskies in Match 1

2.2 Sequence Network

Cooperation patterns between players are represented by motifs in the network. If we look at subgraphs of Whole-match Network, there may be a long time interval between passings in a motif, which means we cannot treat them as a whole. In fact, the ball passing process of a soccer game consists of a number of sequences which represent different attack trials of the team. A way to make a deeper insight into the whole match network needs to be found.

In order to solve this sort of complexity, we define the sequence to see the process more accurately. We are given series of discrete data which represents who is holding the ball before and after each ball passing. Obviously, successive ball passing should have an end to end property. Thus, our sequence is the set of a series of successive ball passing. Once the property is not satisfied, which means an event such as a fuel happens, the sequence will break spontaneously. In this way we get the Sequence Network for a single sequence.

As is shown in Fig.4, we use the arrow to represent the ball passing from the start to the end. The number in the middle of arrow shows the order that the corresponding ball passing happens. Once the ball passing process is not successive, a whistle is added to represent an event. In the process like this, we will derive 4 separate sequences which are shown in the right.

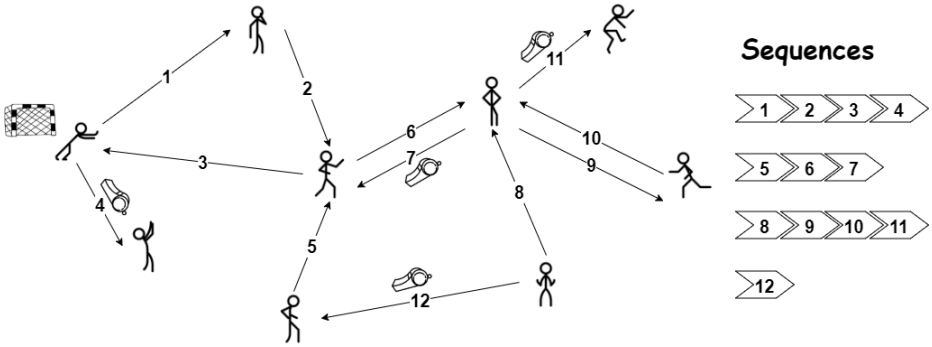

Figure 4: The process to define the sequence

Since the time periods are short, passings in the same sequence can be considered as continuous. In other words, motifs in each Sequence Network represent the cooperation between teammates. Fig.5 shows all possible motifs involving 2 or 3 players.

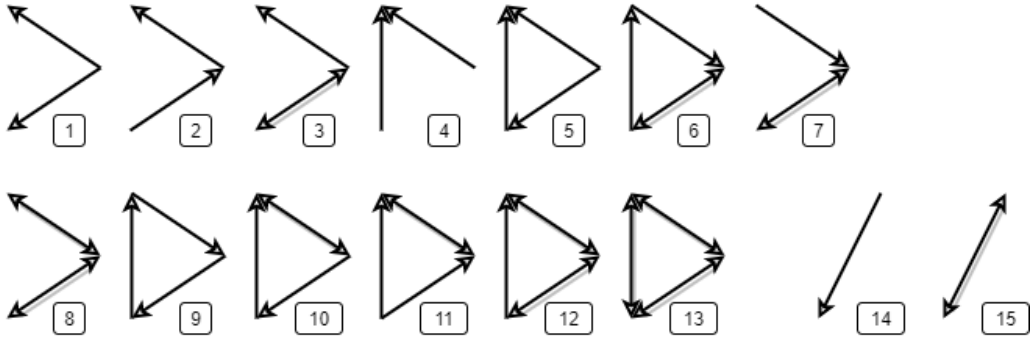

Figure 5: Motifs involving 2 or 3 nodes

We find these motifs extremely helpful in team-level evaluation. Contributions of each motif are discussed in the next section.

3 Performance Indicators

Performance evaluation is extremely difficult for soccer. Apart from the overall performance, knowledge of how each player is involved in the passings would offer more targeted advice for training. However, a player may pass a ball to a player who then make an assist and contribute to a goal. Hence contribution of each team member cannot be properly estimated in any evaluation model based only on shootings and assists. Previous team evaluation models mainly try to predict the outcome[8][9] and pay little attention to performance indicators. Such evaluations help make profit for bookmakers by telling the probability of each outcome, but are too macro for the coach to improve the team.

Based on such nature and our ball passing model, we propose a performance evaluation system which focuses on the performance of the team as well as individual players in each match. With the following performance indicators, we provide a more comprehensive idea of the characteristics of each player and the quality of their performance as a whole team.

3.1 Individual Player Performance Indicator

It is hard to evaluate each passing in the soccer match. Contribution of individual players is more accurately assessed from local network invariant. We employ the following three centrality measures as indicators for how well the player is connected, how much the player helps connection between other teammates, and how popular the player is among his teammates.

3.1.1 Closeness Centrality

In network theory, closeness centrality for a node is

$$c_i = \frac{N}{\sum_{j \neq i} d_{ij} + \sum_{j \neq i} d_{ji}} \tag{1}$$

where d_{ij} is the geodesic distance from player i to player, and N is the number of players involved in the match. Closeness centrality measures the mean distance from a node to other nodes[10]. A high closeness centrality means a smaller mean distance. In a soccer match, larger closeness mirrors better connectivity of a player.

We calculate the average[3] closeness centrality of players in each match and find that the highest by M10. The centrality of M10 in the 38 matches is illustrated in Fig.6[4].

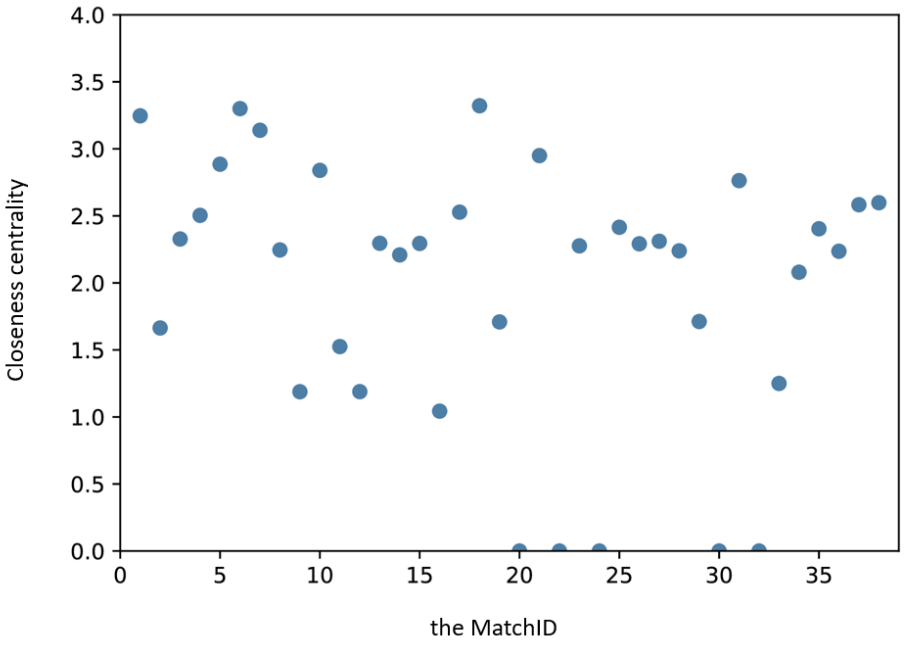

Figure 6: Closeness centrality of M10 in different matchess

The Whole-match Network of the Huskies in Match 1 is illustrated in Fig.3. The size of the node stands for the closeness centrality of the player.

3.1.2 Betweenness Centrality

Betweenness centrality indicates the percentage of shortest paths that go through player i and is defined as

$$C_B(i) = \sum_{j \neq k \neq i} \frac{n^i_{jk}}{g_{jk}} \tag{2}$$

where n^i_{jk} is the number of geodesic paths from j to k going through i and g_{jk} is the total number of geodesic paths[10]. Players with larger betweenness centrality are more important in the connection of teammates.

With calculation, we find that D10 has the largest average betweenness centrality while D1 have the smallest. This indicator helps find the most relied players in a match.

[3]Matches that the player did not attend are not considered while calculating the average.

[4]0 means M10 did not take part in the match.

3.1.3 Pagerank Centrality

Pagerank centrality, defined as

$$x_i = p \sum_{j \neq i} \frac{a_{ji}}{L_j^{out}} x_j + q$$

is introduced by S.Brin et al. to evaluate the relevance and significance of a website[11]. $L_j^{out} = \sum_k a_{ji}$ is the total number of passings made by player j. p and q are heuristic parameters. Here we employ this measurement to evaluate the 'popularity' under the principle that a player who gets passings from his teammates is popular.

3.2 Team Performance Indicator

3.2.1 Motif Contribution Score

As illustrated in Fig.5, there could be several patterns among the continuous passings between players. For each motif, we study the distance and the number of duels during each occurrence. The data is processed as below.

1. For each match, we cut the passings into different sequences based on events such as shoot or foul. We then construct Sequence Networks with each sequence of passings.

2. We find the motifs in each Sequence Network, and record the displacement and duels during each motif.

3. For each motif, mean value of displacement and duels are calculated respectively to measure the contribution of each motif.

In this way we get the average displacement and number of duels for each motif. The calculated data is as in Tab.1. A larger (positive) displacement means after these passings the ball is more likely to be closer to the goal. A larger value of $\frac{duel}{dis}$ during a motif means the rivals struggle to catch the ball but the ball was kept, and thus the motif is less fragile. The occurrences of different motifs is a good indicator for team performance. A very basic idea is to provide a **motif contribution score** which is defined as

$$s = \sum_{i=1}^{15} k_i \frac{n_i}{n} (\alpha \cdot dis_i + \beta \cdot \frac{duel_i}{dis_i})$$

where n_i is the total occurrences of motif i in a match, k_i is the effective coefficient demonstrating the importance of the motif, and n is the total number of motifs in the whole match. From the data, we would put that $\alpha = 8.8866$ and $\beta = 0.7666$. A negative k_i means negative contribution. The outcome (the Huskies win, the Huskies lose, tie) can be represented by the difference of scores between the Huskies and the opponents. It can be calculated as the following formula.

	total occurrences	displacement	number of duels
motif 1	703	19.69	2.62
motif 2	1939	17.04	2.54
motif 3	579	19.57	2.62
motif 4	679	19.14	2.62
motif 5	175	18.87	2.62
motif 6	58	17.28	2.59
motif 7	571	19.19	2.61
motif 8	177	20.69	2.69
motif 9	386	19.17	2.76
motif 10	133	18.62	2.64
motif 11	48	15.83	2.85
motif 12	40	15.78	2.80
motif 13	9	14.78	1.89
motif 14	3366	13.39	2.33
motif 15	786	17.16	2.59

Table 1: Measures for the motifs

$$
\begin{aligned}
p &= \tanh(s_H - s_O) \\
&= \tanh(\sum_{i=1}^{15} k_i \frac{n_{Hi}}{n}(\alpha \cdot dis_i + \beta \cdot \frac{duel_i}{dis_i}) - \sum_{i=1}^{15} k_i \frac{n_{Oi}}{n}(\alpha \cdot dis_i + \beta \cdot \frac{duel_i}{dis_i})) \\
&= \tanh(\sum_{i=1}^{15} k_i \frac{n_{Hi} - n_{Oi}}{n} \cdot (\alpha \cdot dis_i + \beta \cdot \frac{duel_i}{dis_i})).
\end{aligned}
$$

The tanh function is used to converge the value to [-1, 1]. A negative p means the Huskies is likely to lose while a positive p means the Huskies is likely to win. We learned from the data of the 38 matches and the coefficients are as in Tab.2[5].

Motif ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
k	4.37	1.36	6.63	2.59	2.52	33.41	8.04	10.99	3.17	1.75
total occurrences	703	1939	579	679	175	58	571	177	386	133

Motif ID	**11**	**12**	**13**	**14**	**15**
k	16.72	10.19	25.39	0.06	1.60
total occurrences	48	40	9	3366	786

Table 2: Coefficients and occurrences of motifs

With the coefficients in Tab.2, we can cal calculate the difference between motif contribution scores

[5]The red coefficients are negative.

of the two teams[6]. As is shown in Tab.3, a higher motif contribution score is indeed an indication of better team performance. Difference for the winning matches are always positive, indicating that the Huskies are superior. Difference for the losing matches, to the contrary, are always negative. For the ties, however, the difference is sometimes confusing. It is probably because prediction of a tie is too difficult.

MatchID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Outcome	win	tie	lose	lose	lose	win	lose	tie	lose	lose
Prediction	1.00	-0.08	-1.00	-1.00	-1.00	1.00	-1.00	1.00	-1.00	-0.99
MatchID	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Outcome	win	tie	lose	win	win	tie	win	win	tie	tie
Prediction	0.97	-0.11	-1.00	0.98	0.97	-1.00	1.00	0.99	-0.06	1.00
MatchID	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
Outcome	lose	lose	lose	tie	win	lose	win	lose	lose	win
Prediction	-1.00	-1.00	-0.98	1.00	0.99	-1.00	1.00	-0.96	-1.00	0.97
MatchID	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**		
Outcome	win	lose	tie	tie	win	win	tie	lose		
Prediction	0.97	-1.00	-1.00	1.00	1.00	1.00	0.02	-0.98		

Table 3: Outcomes and differences

3.2.2 Clustering Coefficients

The clustering coefficient measures the extent to which the nodes in a network tend to cluster. The modified notion of clustering coefficients[12]

$$c_i^w = \frac{1}{u_i(u_i-1)} \sum_{j,k} \frac{\sqrt[3]{a_{ij}a_{kj}a_{ki}}}{\max(A)}$$

can be used to calculate the clustering coefficient of the Whole-match Network. With the clustering coefficients of different matches, the coach can have a better knowledge of which set of players have better connection and whether the team need more training to enhance the bond between players.

4 Effective Strategies

Based on the insights we gain from the teamwork model, we provide suggestions from the perspective of player cultivation, lineup sequence and cooperation training. The three indicators reflect a diversity of aspects. Some players are better connected to other teammates. Some are essential for connections of other teammates. Some are trusted by their teammates and heavily relied on during the matches.

[6]Equation for the calculation is mentioned above.

4.1 Player Cultivation

From the indicators we find that there is huge difference between players. Since the different centrality measures the relating dimension of ability of a certain player, we could clearly see the individual capacity of the players in this way. An interesting finding is that the referee have the preference to arrange the player with higher individual ability to take part in the soccer match, which consequently leads to a situation that the better a player is, the more practice experience will he gain due to a higher probability to appear in the field. Hence, the team will have a tendency to be unsymmetrical as the strong players will be stronger while the weak player tends to be more raw with few entry opportunities.

Our goal is to design a more effective soccer team, which requires us to thoroughly increase the capacity of the team. Thus, the player with low capacity should also be put enough emphasis on. Player cultivation is proposed to build a more comprehensive soccer team. The referee should consider giving more appear opportunities to the ordinary players. This will help them gain experience and build their individual capability, which in turn leads to a more comprehensive soccer team.

4.2 Lineup Selection

From the data we find that some lineup are not so effective. Attendance of some promising players is not so satisfying. In order to achieve better performance, apart from player cultivation, we would suggest that the couch select the most promising players according to their individual indicator.

Besides, some set players tend to have better cooperation than other teammates. Such phenomenon is obvious with the passings. Attendance of those players in the same team would be a benefit. If the couch could pay more attention to the selection of lineup, the clustering coefficient is more likely to increase.

4.3 Cooperation Training

Using heat-map, we clearly find that Huskies is more likely to win the match when having multiple motifs in the game. For example, in the match 15, the heat-map(Fig.7) shows obvious dynamics of the motifs that are utilizing, which, consequently, lead the Huskies to win the game. The very initial idea is the team should try different patterns in order to become more flexible in the match. Looking at motifs in the matches that the Huskies lost, it is obvious that they tend to stick to some particular dyatic configurations. They would need to focus more on the diversity of their cooperation. Fig.8 is an example from one of the matches that the Huskies lost.

Furthermore, according to Tab.2, it may be more effective to perform certain motifs more frequently. For example, motif 14 is not so beneficial for the success of a match, but it occurred the most frequently. While training for the next match, we would suggest the players to practice patterns represented by motif 6 or motif 8.

5 Generalization to Other Team Work

Based on the previous analysis and modeling, we are now capable to analyze the group dynamics in a controlled setting of the soccer. An analogy could be proposed in order to expand our model in other team cooperation. Some researches on the network between team members except soccer

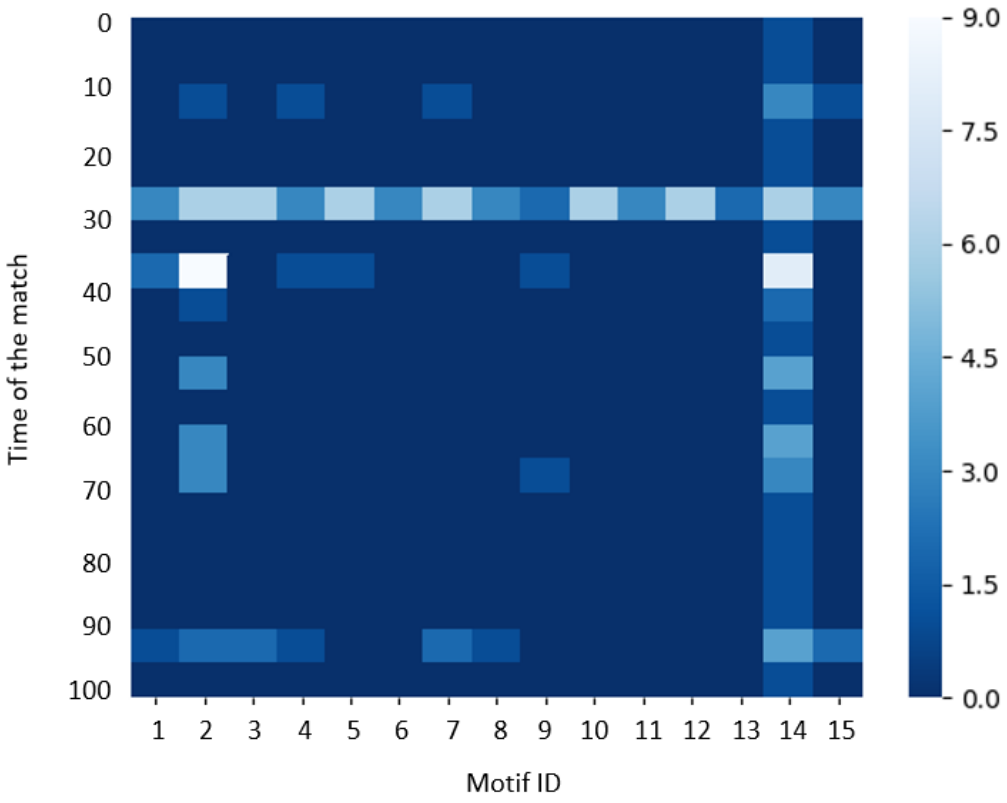

Figure 7: The heat-map for the motifs used by Huskies in Match 15

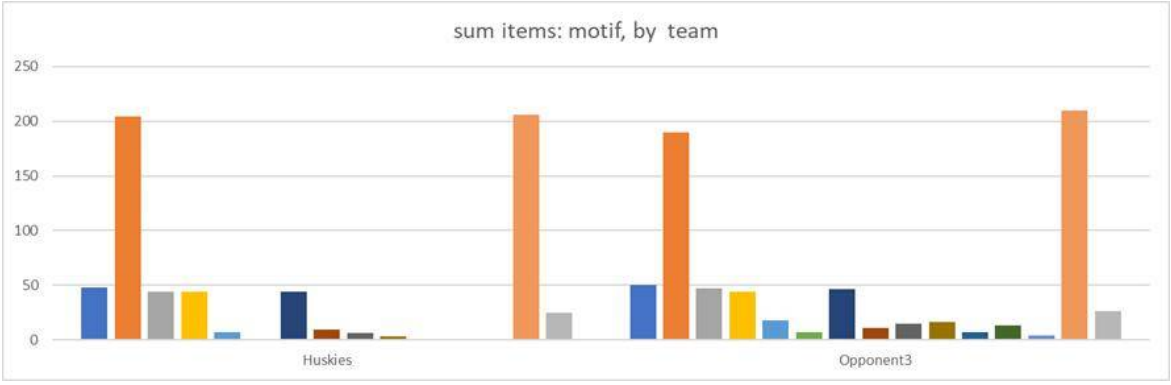

Figure 8: Comparison between the Huskies and the rival in a match that the Huskies lost

have been carried out. Lurie and Yotam have already found that effective interaction between team members could also make contributions to the whole team's performance.[7] It could be concluded that a communication network could be generated out of the passing network in order to mirror the connection between the team members.

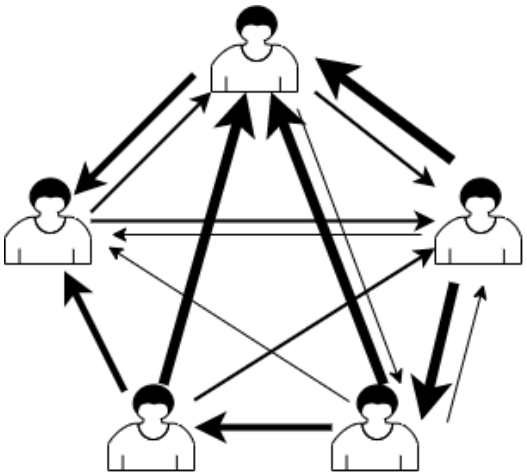

Figure 9: The communication network in a team

5.1 Modifying The Former Model

As to modify the passing network model into the communicating network model, some parameters need to be redefined. For the weight of the communication network, it will be an index that represents the communication between members will be used as the weight of the network. For example, in a team network generated in an office,it may be the number of e-mails that two members have sent to each other in a certain period to represent the extent that they communicate with each other. The node of the communication network remains to be the member of the team. In this way, we could tie the communicating network during a certain period as Fig.9.

Further, the team members communicate with each other in various patterns, which will be shown as motifs to achieve the macro goal of the whole team.

5.2 Evaluate The Communication Network

For the purpose of evaluating the team performance according to different communication networks, both micro and macro aspects will be considered in in order to give a thorough insight into the team performance.

5.2.1 Micro Aspect: Evaluate the individual performance

Since the team is constituted by different members, figuring out the contribution of the team member to the whole team is the base on which to evaluate the effectiveness of the communication network. In other words, we will firstly define some parameters to confirm the certain status of a node in the communication network, which is similar to the passing network with a few differences.

The closeness centrality of the node is defined as the inverse of the average geodesic distance of that node in the network. The closeness centrality provides a direct measurement on how easy it is to reach a particular member within a team. A high closeness centrality corresponds to a small average distance, indicating a well-connected member within the team. This implies that the members with higher closeness centrality have more passion to communicate with others in the team cooperation.

The betweenness centrality measures the extent to which a team member lies on paths between other members. This indicate the status of importance of a member in the communicate network in that the member with high betweenness centrality have a high proboblity to be in a nuclear position of the network to convey information.

The pagerank centrality follows the principle that 'a member is popular if he gets information from other popular members'. This could help to find the node which most messages tend to converge to it. Further, the member that this node represents should be the person that grasp the whole pace of the team.

5.2.2 Macro Aspect: Evaluate the team performance

Various dimensions are considered altogether to reach a more comprehensive standard which grade the performance of a team. For teams that are doing group sports, the key evaluation criterion should

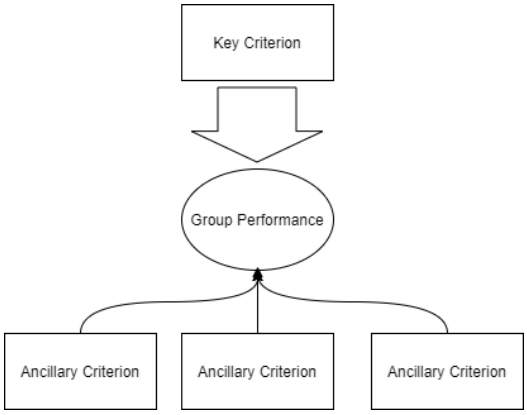

Figure 10: The evaluation standards of the team performance

be the scores that they get during a certain period. Other indexes such as ball control rate, Violation rate could be used as the ancillary grading criterion. As for the teams in a broader sense, similarly, the main criterion of the team should be a quantitative index that mirrors the achievement a team reach in a certain period. Other ancillary criterion will be discussed in the Part 5.4.

5.3 Strategies To Design More Effective team

5.3.1 The Personnel Arrangement of An Effective Team

According to the current theory, the specialized position of every team member have been well defined in the theory of management.[13] However,given the certain position, which member in the team is more suitable for this position? Researches based on the network theory haven't been done to solve

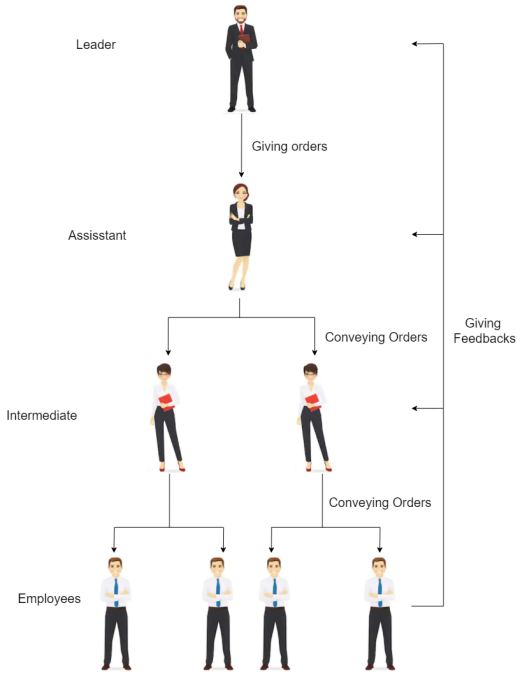

Figure 11: The structure of a team

this problems. Thus, based on the current theory of the management and the network theory, we define the structure of a team as Fig.11.

Using the statistics of the communication between the members in the team, we make the suggestion as to the position of different members in the team as following.

Leader: A team leader should have the strong situation view, which requires the leader to always master the process that the team are undergoing. Therefore, massages will converge to the leader. In this aspect, we use the pagerank centrality to be the key criteria that contribute to the grade that a member could be a leader.

Assistant of the leader: The assistant of the leader should have a strong capability to convey the leader's orders to other members in the team. This represents how easy the member could communicate with others in the team. Thus, we use the closeness centrality as the main factor that contributes to the assistant.

Intermediate: The intermediate is the member in the team who have the contribution when communication between certain members shall happen. Thus, the member with higher betweenness centrality will be more considered as an intermediate in the team.

Employees: The employees consist the most of the members in the team. The main responsibility of them is to execute the orders they received and give feedback to the corresponding members in the team.

5.3.2 Typical Motifs Of An Effective Team

Just like the certain motifs could bring the team with more benefits in the soccer team, certain communicating motifs could also be found in other types of teams. From this perspective, according to the property of the certain team, we could custom a suitable index to measure the outcome of the team.

Using machine learning, we could find the most effective motif to a certain team. The motif could be both 2-node or 3-node. An instance of the motif is shown in the Fig.12.

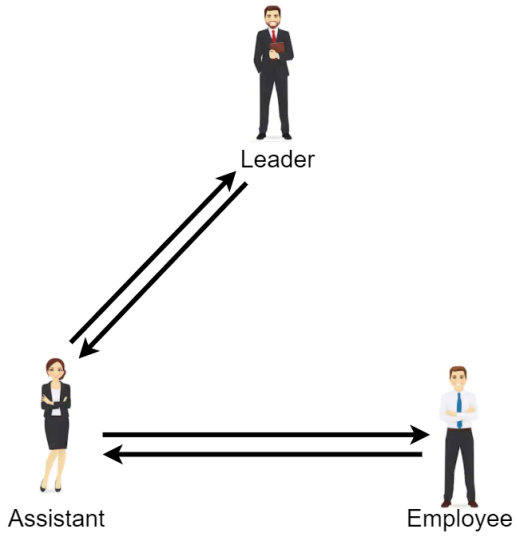

Figure 12: An example for a motif in a team

5.3.3 Clusters In An Effective Team

As mentioned before, we have defined the cluster coefficient to mirror the tendency that a member tends to be a central in a sub-group of the team. The members with higher cluster coefficient should be taken into consideration by the leader in that in a certain sub-group, they will have a crucial status in the certain group. Therefore, they could be seen as the subchief of the small group. Conveying messages to the subchief will be a more effective way for the whole team.

5.4 More Aspects To Capture For An Effective Team

The former analysis are based on the communication level in a team in order to design an effective team. Although the achievement of a team contribute a lot to the performance, other factors that represent the teams' performance could also be taken into consideration to make a more accurate gradescore of the performance of a certain team. Based on the theory of management, the ancillary standards of an effective team could be the happiness that the members feel in the team, the extent that the members communicate with each other in the team and so on.

6 Reflection of the Model

6.1 Strengths

6.1.1 Accuracy of the Motif Contribution Score

The motif contribution score based on our network model could be used to predict the final outcome of a soccer match according to the motifs that a team uses to take part in the match. The simulation result

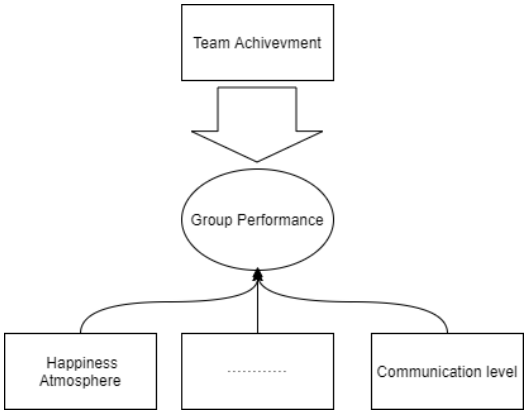

Figure 13: An example for evaluating performance of a certain team

of the machine learning shows a great ability of prediction of the outcome with a relatively lower loss.

6.1.2 Multiple Dimensions to Evaluate a Player

As a player in a soccer team may have different characteristics. It will be hard to formalize the individual capacity of a certain player. We solve this problem by defining three different centrality of the passing network in order to measure different dimensions of the player's ability.

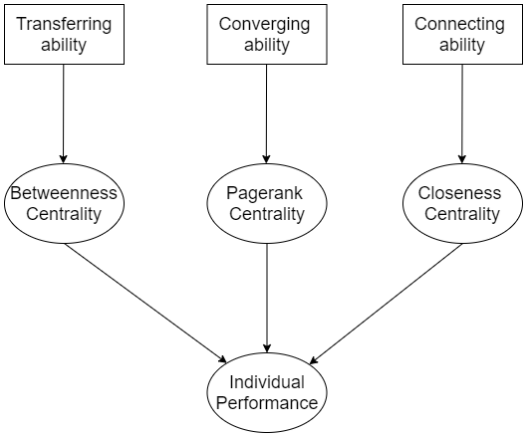

Figure 14: The multiple dimensions to grade a player

With these three indicators, we offer a more comprehensive idea of each team member.

6.1.3 Generalization Ability of the Model

Our model is not only suitable to analyze the performance of a soccer team in a match. With simple modification, it could also be used in the analysis of other types of team work. Based on the theory of the team management, we employ the network theory to provide a distinctive insight into the management of the team. Our model is suitable for quantification of performance in a diversity of teamwork.

6.1.4 The visual analysis of the model

We create a certain network that can visually represent the cooperation between the team members and hence makes it easier for the leader to have an idea of the working status of the whole team. Furthermore, the network helps the decision maker to make more targeted arrangement to make the team more effective. There can be more policies based on the experience of the manager even if the manager is not a statistic scientist.

6.2 Weaknesses and Potential Future Work

6.2.1 Enlarge Soccer Game Dataset

We only have data for 38 match, and opponent teams only have data for 2 matches each. If we could include more soccer matches and learn from data of more teams, our model will have a much better performance.

6.2.2 Synthesis of Indicators of Individual Performance

The three indicators we provide respectively represent three aspects of individual performance. We would like to synthesize the three indicators in order to have a more quantified idea of the overall performance of an individual player. Currently we do not have enough data for some player who only attended 4 or 5 matches. If given more data, we would like to learn how those three indicators can be combined.

6.2.3 Learning from Other Types of Teamwork

We have stated how our model can be applied to other types of teamwork. However, we do not have enough data for other teamwork. In fact, multiple data to measure the team performance is needed such as the communicating level of the whole team, the number of conflicts between the team members should be taken into consideration.

References

[1] Claudio Lucifora. Superstar effects in sport: Evidence from italian soccer. *Journal of Sports Economics*, 4:35–55, 02 2003.

[2] John Whitfield. Collaboration: Group theory. *Nature*, 455:720–3, 11 2008.

[3] Roger Guimerà, Brian Uzzi, Jarrett Spiro, and Luís Amaral. Team assembly mechanisms determine collaboration network structure and team performance. *Science (New York, N.Y.)*, 308:697–702, 05 2005.

[4] Javier Peña and Hugo Touchette. A network theory analysis of football strategies. 06 2012.

[5] Necmi GÜRSAKAL, Firat Yilmaz, Halil Orbay Çobanoğlu, and Sandy Cagliyor. Network motifs in football. 20:263–272, 12 2018.

[6] Qinghe Jing, Weiyan Wang, Junxue Zhang, Han Tian, and Kai Chen. Quantifying the performance of federated transfer learning, 12 2019.

[7] Yotam Lurie. The ethics of cooperation in business. *Open Journal of Philosophy*, 6, 05 2016.

[8] Anthony Constantinou, Norman Fenton, and Martin Neil. pi-football: A bayesian network model for forecasting association football match outcomes. knowledge-based systems, 36, 322-339. *Knowledge-Based Systems*, 36:332–339, 12 2012.

[9] Stefan Samba. *Football Result Prediction by Deep Learning Algorithms*. PhD thesis, 05 2019.

[10] M. Newman. *Networks: An Introduction*. OUP Oxford, 2010.

[11] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117.

[12] Jari Saramäki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and János Kertész. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2), Feb 2007.

[13] Mykolay Chubenko and Dmytro Bedrii. Models of team management in it projects. *Zeszyty Naukowe Wyższej Szkoły Humanitas Zarządzanie*, 19:127–136, 12 2018.

A Source Code

```
"""
the program is served as tuning the parameters of motif indicator
"""

import pandas as pd
import numpy as np
import torch
from torch import nn


class MotifGrade(nn.Module):
    def __init__(self):
        super().__init__()
        self.para = torch.DoubleTensor(np.random.random((2, 1)))
        self.para2 = torch.DoubleTensor(np.random.random((15, 1)))
        self.para.requires_grad = True
        self.para2.requires_grad = True

    def forward(self, x1, x2):
        tt = x2 * self.para2
        x3 = torch.matmul(x1, tt)
        x4 = torch.matmul(x3, self.para)
        x5 = torch.tanh(x4)
        return x5

if __name__ == '__main__':
    base_in = './output/grade/'
    x1 = torch.tensor(np.array(pd.read_csv(base_in + 'trainX1.csv', index_col=0)))
    x2 = torch.tensor(np.array(pd.read_csv(base_in + 'motif_cnt2.csv', index_col=0)))
    y = torch.tensor(np.array(pd.read_csv(base_in + 'trainY.csv', index_col=0)))
    # y -- represent the outcome, win -> 1, loss -> -1, tie -> 0, since the three
        outcomes are not standing alone

    steps = 10000000

    motif_grade = MotifGrade()
    optimizer = torch.optim.SGD([motif_grade.para, motif_grade.para2], lr=2.5e-2)
    loss_fn = nn.MSELoss()

    for step in range(1, steps + 1):
        grade = motif_grade(x1, x2)
        loss = loss_fn(grade, y)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
```

```
        if step % 10000 == 0:
            print("Step: [%d] Loss: %F" % (step, loss.item()))
            print(motif_grade.para.T)
            print(motif_grade.para2.T)
            print(grade.T)
```

```
"""
to integrate data relevant of motif
"""
import pandas as pd

if __name__ == '__main__':
    base_in = './output/motif/motif_'
    match_in = './data/matches.csv'
    base_out = './output/analysis/Match_'
    eval_dir = './output/analysis/score_motif.csv'
    eval = pd.read_csv(eval_dir, header=0, index_col=0)
    matches = pd.read_csv(match_in, index_col=0)
    col = []
    for i in [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15]:
        col.append('motif_' + str(i))
    ret = pd.DataFrame(columns=col + ['coach', 'side', 'scores', 'final_scores',
        'outcome'])
    for ID in range(1, 39):
        mo1 = pd.read_csv(base_in + str(ID) + '_1H.csv', header=0, index_col=0)
        mo2 = pd.read_csv(base_in + str(ID) + '_2H.csv', header=0, index_col=0)
        mo = pd.concat([mo1, mo2], axis=0).loc[:, ]
        husk_idx = (mo['team'] == "Huskies")
        oppo_idx = ~husk_idx
        count_motif = pd.DataFrame(columns=col)
        for i in [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15]:
            pre_m = 'motif_' + str(i)
            mo_idx = (mo[pre_m] > 0)
            count_motif.loc['Huskies', pre_m] = sum(mo_idx & husk_idx)
            count_motif.loc['Opponent', pre_m] = sum(mo_idx & oppo_idx)
        ans = pd.DataFrame(columns=col + ['coach', 'side', 'scores', 'final_scores',
            'outcome'])
        ans.loc['Huskies', 'scores'] = 0
        ans.loc['Opponent', 'scores'] = 0
        for i in [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 15]:
            pre_m = 'motif_' + str(i)
            ans.loc['Huskies', pre_m] = count_motif.loc['Huskies', pre_m] /
                sum(count_motif.loc['Huskies'])
            ans.loc['Opponent', pre_m] = count_motif.loc['Opponent', pre_m] /
                sum(count_motif.loc['Opponent'])
            ans.loc['Huskies', 'scores'] += (ans.loc['Huskies', pre_m] *
                eval.loc[pre_m, 'score'])
```

```
            ans.loc['Opponent', 'scores'] += (ans.loc['Opponent', pre_m] *
                eval.loc[pre_m, 'score'])
        ans.loc['Huskies', 'final_scores'] = ans.loc['Huskies', 'scores'] *
            sum(count_motif.loc['Huskies'])
        ans.loc['Opponent', 'final_scores'] = ans.loc['Opponent', 'scores'] *
            sum(count_motif.loc['Opponent'])
        ans.loc['Huskies', 'coach'] = matches.loc[ID, 'CoachID']
        ans.loc['Huskies', 'side'] = matches.loc[ID, 'Side']
        ans.loc['Huskies', 'outcome'] = matches.loc[ID, 'Outcome']
        ret = pd.concat([ret, ans], axis=0)
        ans.to_csv(base_out + str(ID) + '.csv')
    ret.to_csv(base_out + 'sum.csv')
```

```
"""
the program is used to draw the passing network applying networkx
"""
import networkx as nx
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt


if __name__ == '__main__':
    pass_dir = './image/Match1.csv'
    pos_dir = './image/pos_1.csv'
    close_dir = './image/closeness.csv'
    pass_csv = pd.read_csv(pass_dir, header=0, index_col=0)
    pos_csv = pd.read_csv(pos_dir, header=0, index_col=0)
    close_csv = pd.read_csv(close_dir, header=0)
    player_name = close_csv.columns.to_list()
    pos = {}
    for i in range(11):
        pos[player_name[i]] = (pos_csv.iloc[i, 0], pos_csv.iloc[i, 1])

    G = nx.DiGraph()
    for i in range(11):
        for j in range(11):
            G.add_edge(player_name[i], player_name[j])
    close_list = close_csv.iloc[0].to_list()
    pass_net = []
    for i in range(11):
        pass_net += pass_csv.iloc[i].to_list()

    node_sizes = [i * 120 for i in close_list] # node_sizes represents the closeness
        of the players
    edge_colors = [i + 20 for i in pass_net] # use the color exchange to note the
        passing times
```

```
    edge_alphas = [0 if i == 0 else 1 for i in pass_net]
    # if there's no passing between the certain two player, the edge is invisible
    # edge_width = [i / 2 for i in pass_net]
    edge_width = 6

    nodes = nx.draw_networkx_nodes(G, pos, node_size=node_sizes, node_color='red')
    edges = nx.draw_networkx_edges(G, pos, node_size=node_sizes, arrowstyle='->',
                                   arrowsize=20, edge_color=edge_colors,
                                   edge_cmap=plt.cm.Blues, width=edge_width)
    # set alpha value for each edge
    for i in range(11):
        edges[i].set_alpha(edge_alphas[i])

    pc = mpl.collections.PatchCollection(edges, cmap=plt.cm.Blues)
    pc.set_array(pass_net)
    plt.colorbar(pc)

    ax = plt.gca()
    ax.set_axis_off()
    plt.show()
```

```
import pandas as pd


if __name__ == '__main__':
    base_dir = './output/motif/motif_'
    col = ['team', 'duel', 'dis']
    for i in range(1, 16):
        col.append('motif_' + str(i))

    output_base_dir = './output/analysis/score_motif.csv'
    mo = pd.DataFrame(columns=col)
    ans = pd.DataFrame(columns=['times', 'dis', 'duel'])
    for ID in range(1, 39):
        mo1 = pd.read_csv(base_dir + str(ID) + '_1H.csv', usecols=col)
        mo2 = pd.read_csv(base_dir + str(ID) + '_2H.csv', usecols=col)
        mo = pd.concat([mo, mo1, mo2], axis=0)
    husk_idx = (mo['team'] == 'Huskies')
    scene = mo.loc[husk_idx]
    for i in range(1, 16):
        pre_m = 'motif_' + str(i)
        mo_idx = (scene[pre_m] > 0)
        ans.loc[pre_m, 'times'] = sum(mo_idx)
        ans.loc[pre_m, 'dis'] = sum(scene.loc[mo_idx, 'dis']) / sum(mo_idx)
        ans.loc[pre_m, 'duel'] = sum(scene.loc[mo_idx, 'duel']) / sum(mo_idx)
        ans.loc[pre_m, 'score'] = ans.loc[pre_m, 'dis'] * ans.loc[pre_m, 'duel']
```

```
    ans.to_csv(output_base_dir)
```

```
"""
the program is used to divide sequences in matches,
record some valuable information in fullevents.csv and matches.csv,
and then count the motifs

the sequences are divided according to the possession of the ball
"""

import pandas as pd
from itertools import permutations
import json


def stat_motif(pass_mat):
    motif_mat = [0 for _ in range(16)]
    p_num = len(pass_mat)
    if p_num < 3:
        if p_num >= 2:
            for p in permutations(range(p_num), 2):
                motif_mat[14] += pass_mat.iloc[p[0], p[1]]
                motif_mat[15] += min(pass_mat.iloc[p[0], p[1]], pass_mat.iloc[p[1],
                    p[0]])
            motif_mat[15] /= 2
            # motif_mat[14] -= motif_mat[15]
        return motif_mat

    # count all motifs
    for p in permutations(range(p_num), 3):
        # choose all possible three players i.e. the permutations
        # and then check that whether the sequence meet some formation conditions of
            the motifs
        motif_mat[1] += min(pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[1], p[2]])
        motif_mat[2] += min(pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[2], p[1]])
        motif_mat[3] += min(pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[1], p[2]],
            pass_mat.iloc[p[2], p[1]])
        motif_mat[4] += min(pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[2], p[0]])
        motif_mat[5] += min(pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[2], p[0]],
            pass_mat.iloc[p[1], p[2]])
        motif_mat[6] += min(pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[2], p[0]],
            pass_mat.iloc[p[1], p[2]],
                        pass_mat.iloc[p[2], p[1]])
        motif_mat[7] += min(pass_mat.iloc[p[0], p[1]], pass_mat.iloc[p[1], p[2]],
            pass_mat.iloc[p[2], p[1]])
        motif_mat[8] += min(pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[0], p[1]],
            pass_mat.iloc[p[1], p[2]],
```

```
                        pass_mat.iloc[p[2], p[1]])
        motif_mat[9] += min(pass_mat.iloc[p[0], p[1]], pass_mat.iloc[p[1], p[2]],
            pass_mat.iloc[p[2], p[0]])
        motif_mat[10] += min(pass_mat.iloc[p[0], p[1]], pass_mat.iloc[p[1], p[2]],
            pass_mat.iloc[p[2], p[0]],
                        pass_mat.iloc[p[1], p[0]])
        motif_mat[11] += min(pass_mat.iloc[p[0], p[1]], pass_mat.iloc[p[2], p[1]],
            pass_mat.iloc[p[2], p[0]],
                        pass_mat.iloc[p[1], p[0]])
        motif_mat[12] += min(pass_mat.iloc[p[0], p[1]], pass_mat.iloc[p[1], p[2]],
            pass_mat.iloc[p[2], p[0]],
                        pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[2], p[1]])
        motif_mat[13] += min(pass_mat.iloc[p[0], p[1]], pass_mat.iloc[p[1], p[2]],
            pass_mat.iloc[p[2], p[0]],
                        pass_mat.iloc[p[1], p[0]], pass_mat.iloc[p[2], p[1]],
                            pass_mat.iloc[p[0], p[2]])

    for p in permutations(range(p_num), 2):
        motif_mat[14] += pass_mat.iloc[p[0], p[1]]
        motif_mat[15] += min(pass_mat.iloc[p[0], p[1]], pass_mat.iloc[p[1], p[0]])

    # remove the repeated motif due of the symmetry
    motif_mat[1] /= 2
    motif_mat[4] /= 2
    motif_mat[6] /= 2
    motif_mat[8] /= 2
    motif_mat[9] /= 3
    motif_mat[11] /= 2
    motif_mat[13] /= 3
    motif_mat[15] /= 2

    return motif_mat


def div_seq(seq_csv, full_csv, pd_name, matchID, matchPeriod):
    assert (matchID > 0 & matchID <= 38)
    scene_idx = (seq_csv.loc[:, 'MatchID'] == matchID)
    event_idx = (full_csv.loc[:, 'MatchID'] == matchID)
    output_dir = 'output/motif/motif_' + str(matchID)
    if matchPeriod == '1H' or matchPeriod == '2H':
        scene_idx = scene_idx & (seq_csv.loc[:, 'MatchPeriod'] == matchPeriod)
        event_idx = event_idx & (full_csv.loc[:, 'MatchPeriod'] == matchPeriod)
        output_dir += ('_' + matchPeriod)
    output_dir += '.csv'

    scene = seq_csv.loc[scene_idx].loc[:,
           ['TeamID', 'OriginPlayerID', 'DestinationPlayerID', 'EventTime',
```

```
            'EventOrigin_x', 'EventDestination_x']]
scene.index = range(sum(scene_idx))
scene.columns = ['Team', 'Orig', 'Dest', 'Time', 'ox', 'dx']

event = full_csv.loc[event_idx].loc[:, ['TeamID', 'EventTime', 'EventType']]
event.index = range(sum(event_idx))
event.columns = ['Team', 'Time', 'Event']
duel_idx = (event.loc[:, 'Event'] == 'Duel')
shot_idx = (event.loc[:, 'Event'] == 'Shot')
# split into two teams to count
hask_idx = (event.loc[:, 'Team'] == 'Huskies')
oppo_idx = ~hask_idx

data_store = []

orig = scene.loc[0, 'Orig']
dest = orig
team = scene.loc[0, 'Team']
s_time = scene.loc[0, 'Time']
ox = scene.loc[0, 'ox']
dx = scene.loc[0, 'dx']
player_inv = set()
player_inv.add(orig)
Pass_mat = pd.DataFrame()
Pass_mat[orig] = 0
Pass_mat.loc[orig] = 0

for index, row in scene.iterrows():
    if row['Orig'] != dest:
        # means that the last sequence ended
        e_time = row['Time']
        Motif_mat = stat_motif(Pass_mat)
        tmp_dict = {}
        tmp_dict['player_inv'] = player_inv
        tmp_dict['motif'] = Motif_mat
        tmp_dict['s_time'] = s_time
        tmp_dict['e_time'] = e_time
        tmp_dict['team'] = team
        tmp_dict['ox'] = ox
        tmp_dict['dx'] = dx
        time_idx = (event.loc[:, 'Time'] >= s_time) & (event.loc[:, 'Time'] <
            e_time)
        if team == 'Huskies':
            tmp_dict['duel'] = sum(time_idx & duel_idx & oppo_idx)
            tmp_dict['shot'] = sum(time_idx & shot_idx & hask_idx)
        else:
            tmp_dict['duel'] = sum(time_idx & duel_idx & hask_idx)
```

```
            tmp_dict['shot'] = sum(time_idx & shot_idx & oppo_idx)
        data_store.append(tmp_dict)

        orig = row['Orig']
        dest = row['Dest']
        team = row['Team']
        s_time = row['Time']
        ox = row['ox']
        dx = row['dx']
        player_inv = set()
        player_inv.add(orig)
        player_inv.add(dest)
        Pass_mat = pd.DataFrame()
        Pass_mat[orig] = 0
        Pass_mat.loc[orig] = 0
        Pass_mat[dest] = 0
        Pass_mat.loc[dest] = 0
        Pass_mat.loc[orig, dest] += 1
    else:
        dest = row['Dest']
        dx = row['dx']
        if dest not in player_inv:
            Pass_mat[dest] = 0
            Pass_mat.loc[dest] = 0
            player_inv.add(row['Dest'])
        Pass_mat.loc[row['Orig'], dest] += 1

e_time = 3900.0 # 3900 is the longest possible time for a match
Motif_mat = stat_motif(Pass_mat)
tmp_dict = {}
tmp_dict['player_inv'] = player_inv
tmp_dict['motif'] = Motif_mat
tmp_dict['s_time'] = s_time
tmp_dict['e_time'] = e_time
tmp_dict['team'] = team
tmp_dict['ox'] = ox
tmp_dict['dx'] = dx
time_idx = (event.loc[:, 'Time'] >= s_time) & (event.loc[:, 'Time'] < e_time)
tmp_dict['duel'] = sum(time_idx & duel_idx)
tmp_dict['shot'] = sum(time_idx & shot_idx)
data_store.append(tmp_dict)

data_pd = pd.DataFrame(columns=pd_name)
for idx, li in enumerate(data_store):
    data_pd.loc[idx, 'team'] = li['team']
    data_pd.loc[idx, 's_time'] = li['s_time']
    data_pd.loc[idx, 'e_time'] = li['e_time']
```

```
        data_pd.loc[idx, 'ox'] = li['ox']
        data_pd.loc[idx, 'dx'] = li['dx']
        data_pd.loc[idx, 'dis'] = li['dx'] - li['ox']
        data_pd.loc[idx, 'duel'] = li['duel']
        data_pd.loc[idx, 'shot'] = li['shot']
        for ix in range(1, 16):
            data_pd.loc[idx, 'motif_' + str(ix)] = li['motif'][ix]
        if li['team'] == 'Huskies':
            for name in li['player_inv']:
                data_pd.loc[idx, name] = True

    data_pd.to_csv(output_dir)


if __name__ == '__main__':
    passing_dir = './data/passingevents.csv'
    full_dir = './data/fullevents.csv'
    name_dir = './data/players.json'
    Seq_csv = pd.read_csv(passing_dir,
                          usecols=['MatchID', 'TeamID', 'OriginPlayerID',
                              'DestinationPlayerID', 'MatchPeriod',
                                  'EventTime', 'EventOrigin_x', 'EventDestination_x'])
    Full_csv = pd.read_csv(full_dir, usecols=['MatchID', 'MatchPeriod', 'EventTime',
        'EventType'])
    with open(name_dir) as f:
        player_name = json.load(f)
    Pd_name = ['team', 's_time', 'e_time', 'duel', 'shot', 'ox', 'dx', 'dis']
    for i in range(1, 16):
        Pd_name.append('motif_' + str(i))
    Pd_name += player_name

    for ID in range(1, 39):
        div_seq(Seq_csv, Full_csv, Pd_name, ID, '1H')
        div_seq(Seq_csv, Full_csv, Pd_name, ID, '2H')
```

```
"""
the program is used to generate the data in need to tune the parameters of motif
    indicator
"""
import pandas as pd

if __name__ == '__main__':
    base_in = './output/motif/motif_'
    match_in = './data/matches.csv'
    base_out = './output/grade/'

    matches = pd.read_csv(match_in, index_col=0)
```

```
    res = pd.DataFrame(columns=['outcome'])
    for i in range(1, 39):
        if matches.loc[i, 'Outcome'] == 'win':
            res.loc[i, 'outcome'] = 1
        else:
            if matches.loc[i, 'Outcome'] == 'loss':
                res.loc[i, 'outcome'] = -1
            else:
                res.loc[i, 'outcome'] = 0
    # res.to_csv(base_out + 'trainY.csv')

    col = []
    for i in range(1, 16):
        col.append('motif_' + str(i))
    ret = pd.DataFrame(columns=col)
    for ID in range(1, 39):
        mo1 = pd.read_csv(base_in + str(ID) + '_1H.csv', header=0, index_col=0)
        mo2 = pd.read_csv(base_in + str(ID) + '_2H.csv', header=0, index_col=0)
        mo = pd.concat([mo1, mo2], axis=0).loc[:, ]
        husk_idx = (mo['team'] == "Huskies")
        oppo_idx = ~husk_idx
        count_motif = pd.DataFrame(columns=col)
        for i in range(1, 16):
            pre_m = 'motif_' + str(i)
            mo_idx = (mo[pre_m] > 0)
            count_motif.loc['Huskies', pre_m] = sum(mo_idx & husk_idx)
            count_motif.loc['Opponent', pre_m] = sum(mo_idx & oppo_idx)
        for i in range(1, 16):
            pre_m = 'motif_' + str(i)
            # ret.loc[ID, pre_m] = count_motif.loc['Huskies', pre_m] /
                sum(count_motif.loc['Huskies']) - count_motif.loc[
            #     'Opponent', pre_m] / sum(count_motif.loc['Opponent'])
            ret.loc[ID, pre_m] = (count_motif.loc['Huskies', pre_m] - count_motif.loc[
                'Opponent', pre_m]) / (sum(count_motif.loc['Opponent']) +
                    sum(count_motif.loc['Huskies']))
    # ret.to_csv(base_out + 'trainX.csv')
    ret.to_csv(base_out + 'trainX1.csv')
```

```
"""
the program is used to calculate the tactical position of the players
the tactical position is the central position of the players appearance position in
    a match
"""
import pandas as pd
import json
```

```
def cal_pos(event, player_name):
    data_pd = pd.DataFrame(columns=['pos_x', 'pos_y'])
    for name in player_name:
        orig_idx = (event['OriginPlayerID'] == name)
        dest_idx = (event['DestinationPlayerID'] == name)
        num = sum(orig_idx) + sum(dest_idx)
        if num == 0:
            continue
        pos_x = (sum(event.loc[orig_idx, 'EventOrigin_x']) + sum(event.loc[dest_idx,
            'EventDestination_x'])) / num
        pos_y = (sum(event.loc[orig_idx, 'EventOrigin_y']) + sum(event.loc[dest_idx,
            'EventDestination_y'])) / num
        data_pd.loc[name, 'pos_x'] = pos_x
        data_pd.loc[name, 'pos_y'] = pos_y
    return data_pd


if __name__ == '__main__':
    full_dir = './data/fullevents.csv'
    name_dir = './data/players.json'
    full_csv = pd.read_csv(full_dir,
                           usecols=['MatchID', 'TeamID', 'OriginPlayerID',
                               'DestinationPlayerID', 'MatchPeriod',
                                   'EventOrigin_x', 'EventOrigin_y',
                                      'EventDestination_x', 'EventDestination_y'])
    full_csv = full_csv.dropna(axis=0, how='any')
    with open(name_dir) as f:
        Player_name = json.load(f)

    for ID in range(1, 39):
        # event_idx = (full_csv.loc[:, 'MatchID'] == ID) & (full_csv.loc[:,
            'MatchPeriod'] == '1H')
        event_idx = (full_csv.loc[:, 'MatchID'] == ID)
        Event = full_csv.loc[event_idx].loc[:,
               ['OriginPlayerID', 'DestinationPlayerID', 'EventOrigin_x',
                   'EventOrigin_y',
                'EventDestination_x', 'EventDestination_y']]
        data_pd = cal_pos(Event, Player_name)
        # output_dir = 'output/pos/pos_' + str(ID) + '_1H.csv'
        output_dir = 'output/pos/pos_' + str(ID) + '.csv'
        data_pd.to_csv(output_dir)

        # event_idx = (full_csv.loc[:, 'MatchID'] == ID) & (full_csv.loc[:,
            'MatchPeriod'] == '2H')
        # Event = full_csv.loc[event_idx].loc[:,
        #        ['OriginPlayerID', 'DestinationPlayerID', 'EventOrigin_x',
            'EventOrigin_y',
```

```
        #         'EventDestination_x', 'EventDestination_y']]
        # data_pd = cal_pos(Event, Player_name)
        # output_dir = 'output/pos/pos_' + str(ID) + '_2H.csv'
        # data_pd.to_csv(output_dir)
```

```
import pandas as pd
import numpy as np
import json

with open('../passing_match/players.json') as f:
    Players = json.load(f)

mat_c = np.zeros((38, len(Players)))

for MatchID in range(1, 39):
    MatD = pd.read_csv('../mat_d/Match'+str(MatchID)+'.csv')
    MatD = np.asarray(MatD)
    w = 0.5
    for i in range(len(Players)):
        sum1 = 0
        sum2 = 0
        for j in range(len(Players)):
            if j == i:
                continue
            if MatD[i][j] < np.inf:
                sum1 += MatD[i][j]
            if MatD[j][i] < np.inf:
                sum2 += MatD[j][i]
        if w * sum1 + (1 - w) * sum2 > 0:
            mat_c[MatchID-1][i] = 13/(w * sum1 + (1 - w) * sum2)
#         else:
#             mat_c[MatchID-1][i] = np.inf

mat_c = pd.DataFrame(mat_c)
mat_c.to_csv('../eval/closeness.csv', header=Players, index=False)
```

```
import pandas as pd
import numpy as np
import networkx as nx
import json

with open('../passing_match/players.json') as f:
    Players = json.load(f)

pagerank = np.zeros((38, len(Players)))
```

```
for MatchID in range(1, 39):
    MatL = pd.read_csv('../mat_l/Match'+str(MatchID)+'.csv')
    MatL = np.asarray(MatL)
    DG = nx.DiGraph()
    for i in range(len(Players)):
        DG.add_node(i)
    for i in range(len(Players)):
        for j in range(len(Players)):
            if i == j:
                continue
            if MatL[i][j] < np.inf:
                DG.add_weighted_edges_from([(i, j, MatL[i][j])])
    pr = nx.pagerank(DG, alpha=0.85)
    for i in range(len(Players)):
        pagerank[MatchID-1][i] = pr[i]

pagerank = pd.DataFrame(pagerank)
pagerank.to_csv('../eval/pagerank.csv', header=Players, index=False)
```

```
import pandas as pd
import numpy as np
import json

with open('../passing_match/players.json') as f:
    Players = json.load(f)

mat_clu = np.zeros((38, len(Players)))

for MatchID in range(1, 39):
    A = pd.read_csv('../passing_match/Match'+str(MatchID)+'.csv')
    A = np.asarray(A)
    MatE = pd.read_csv('../mat_e/Match'+str(MatchID)+'.csv')
    MatE = np.asarray(MatE)
    for i in range(len(Players)):
        if sum(MatE[i]) <= 1:
            continue
        cur = 0
        u = sum(MatE[i])
        for j in range(len(Players)):
            if j == i:
                continue
            for k in range(len(Players)):
                if j == k or i == k:
                    continue
                cur += ((A[i][j]*A[k][j]*A[k][i])**(1.0/3)) / A.max()
        mat_clu[MatchID-1][i] = cur / (u*(u-1))
```

```
mat_clu = pd.DataFrame(mat_clu)
mat_clu.to_csv('../eval/clustering.csv', header=Players, index=False)
```