Problem Chosen	2020 MCM/ICM Summary Sheet	Team Control Number
D		6666666

Go, Huskies!

Teamwork plays an important role in sports competition, such as volleyball, basketball and football. Recently, the coach of Huskies, a football team, has sent their data to us and hopes to gain some insights from us to improve teamwork and better their performance.

For task 1, we model the **Ball Passing Network** and compute topological distance, clustering coefficient and some other related attributes. It's worth mentioning that **Laplace Smoothing Method (LSM)** is introduced to enhance the reliability and stability of our model. By applying network science and graph theory, we develop not only a couple of successful passing patterns but also some decent overall team strategies.

For task 2, we define **capability vector** for each player, with which we draw 5-dimension radar maps (like in FIFA2019). Then, by applying **Poisson Linear Discriminant Analysis (PLDA)**, we modify and rescale those features, thus fitting them well into the measurement of the whole team. And this will guide us to choose wisely from established strategies and targeted ones in front of different opponents.

For task 3, to make sequential decisions for attacks, **Markov Decision Process (MDP)** tuple set is constructed, in which those features collected from previous models are analyzed and manipulated comprehensively to help provide more reasonable tactics to our coach. Besides, we have also pointed out some weakness in defences based on statistical data.

For task 4, we design a series of indicators to measure the success of teamwork. Moreover, it's also workable to generalize these standards to other teams and help enhance their performance.

Finally, we draw conclusions and analyze the sensitivity, strengths and weaknesses of our models.

Key Words: Network Science; Graph Theory; Laplace Smoothing Method; Generalized Clustering Coefficient; Average Path Length; Dijkstra Algorithm; Poisson Linear Discriminant Analysis; Maximum Likelihood Estimate; Markov Decision Process; Group Dynamics

Contents

1 Introduction **4**

1.1 Background 4

1.2 Restatement of the Problem 4

2 Analysis and Approach Overview **5**

3 Assumptions and Notations **6**

3.1 General Assumptions 6

3.2 Notations 6

4 Models **7**

4.1 Football Passing Network 7

4.1.1 Average Path Length 8

4.1.2 Clustering Coefficient 8

4.2 Player Analysis Model 10

4.2.1 Radar Graph 10

4.2.2 The Scoring of Each Player 11

4.2.3 The Scoring of the Whole Team 13

4.3 Tactics for Attacks and Defences 13

4.3.1 Decisions for Attacks 13

4.3.2 Decisions for Defences 17

5 Optimization and Extension for Our Model **18**

5.1 Opinions in Group Dynamics 18

5.2 Discussion on Quaternary Configuration 19

6 Sensitivity Analysis **20**

6.1 Stability of Clustering Coefficient 20
6.2 Stability of Average Path Length 20

7 Strengths and Weaknesses **21**
7.1 Strengths 21
7.2 Weaknesses 22

8 Conclusions **22**

Appendices **24**

Appendix A Markov Decision Process **24**

Appendix B Calculating the Clustering Coefficient **24**

Appendix C Calculating the Average Path Length **27**

Appendix D Proof of Iteration **29**

1 Introduction

1.1 Background

Nowadays, social interconnections are becoming closer and closer. Meanwhile, the challenges we encounter are also becoming much more complex, which remind us of the significance of cooperation. Over the last few decades, diverse methods have been employed to address these tough issues so that teams can work together more efficiently.

In some competitive team sports, team processes become more informative because all team members subject to strict rules. For these teams, their victory is not only decided by the sum of individuals' abilities but also depends on many other factors relating to how well the teammates cooperate together.

The coach of the Huskies is eager to understand his team's dynamics and to know how the complex interactions among the players on the field impacts their success. Mathematical models are introduced in order to help his team plan scientific and specific strategies that can improve teamwork in next season.

1.2 Restatement of the Problem

To fulfill the coach's requirements, his team's features should be analysed thoroughly, while some tasks are to be accomplished:

- Take players on the field as nodes, which are interconnected by passes of the ball, utilize methods in *Network Science* and *Graph Theory* to analyse the feature of them.
- Identify some indicators that can reflect the team's behavior. These indicators can help us to judge the team's performance, to provide guidance for players' strategy and to predict the total rank of the Huskies.
- Based on the model and related indicators, inform the coach about what kinds of structural strategies have been effective for the Huskies, and then advise the coach on what changes his team should make in next season to better their performance.

2 Analysis and Approach Overview

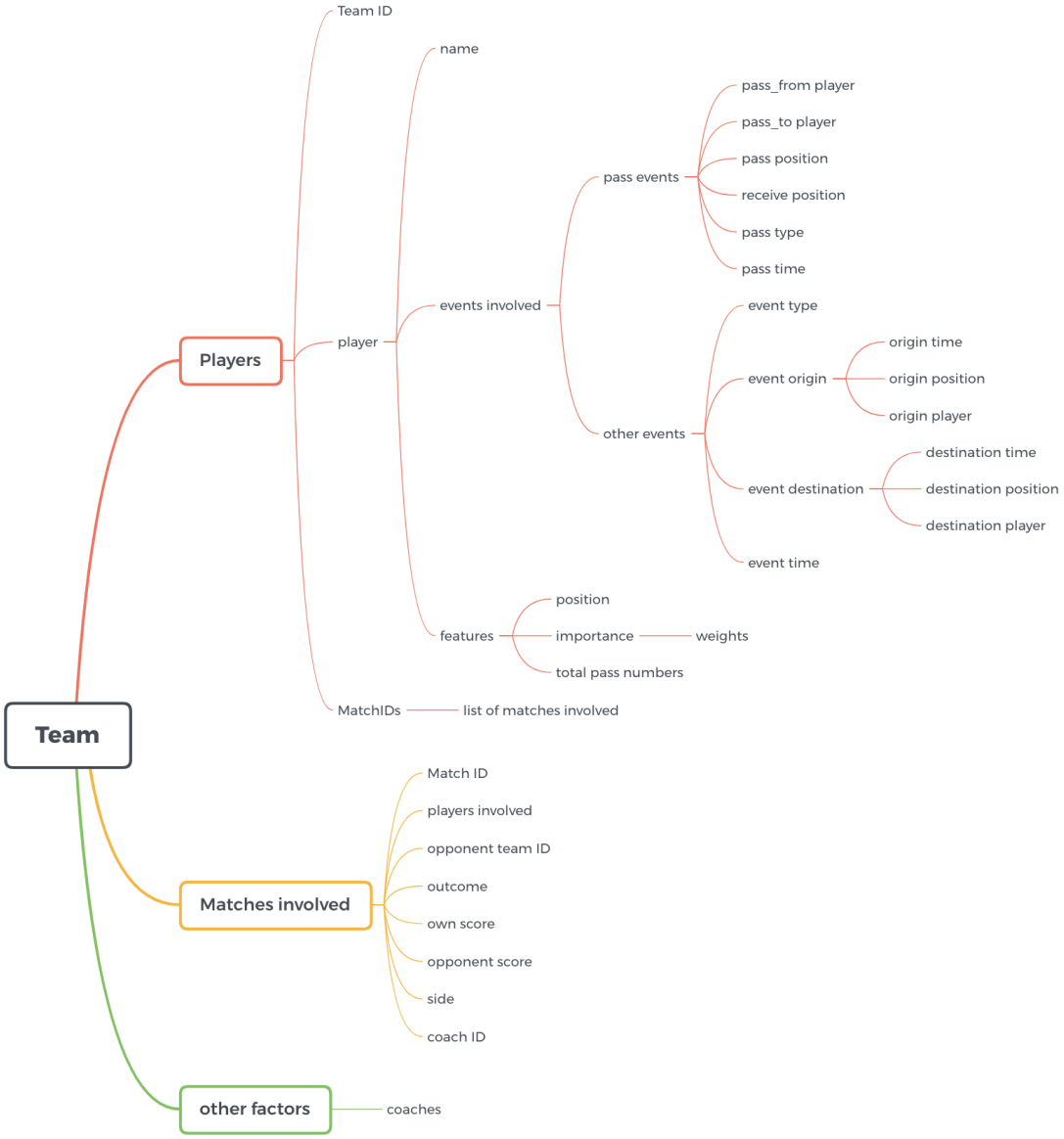


Figure 1: The mind mapping

We explore the data and rearrange the information structure as in Figure 1, and then manipulate these factors as follows:

First, we construct the network model according to passing records for to recognize the passing patterns and some other attributes. Second, we develop personal scoring model and rescale it to measure the score of a team. Third, we arrange attack and defence organization . And finally, some of the universal features are extracted and generalized to fit other kinds of group work as well.

3 Assumptions and Notations

3.1 General Assumptions

- **The graph of passing network in a match is interconnected, which guarantees that the network graph is connected.** Players on the field can't play without receiving or passing balls.
- **Each player on the field plays selflessly only for the victory of his team.** I.e. he tries to maximize the scores, and simultaneously minimize the losses for his team.
- **The referee judges fairly in the match.** Sometimes a team may lose the game because of unfair judgements, but here we mainly focus on a team's strategies and players' performance, thus neglecting those factors of little probability.

3.2 Notations

Notations that we use in the article are listed in the following table:

Notation	Definition
$G(V, E)$	The Graph that represents the Huskies' passing network
$\boldsymbol{A}$	The adjacent matrix of $G(V, E)$
N	The number of nodes in $G(V, E)$
v_i	The ith player on the field, i.e. the ith node in $G(V, E)$
$w_{ij}, \hat{w}_{ij}$	Weight and normalized weight between ith and jth node
l_{ij}	The topology distance between the ith and the jth node
$C_i^{(2)}, C^{(2)}$	The second-order clustering coefficient of v_i and $G(V, E)$
$C_i^{(3)}, C^{(3)}$	The third-order clustering coefficient of v_i and $G(V, E)$
L	The average path length of $G(V, E)$
$R(v_i)$	The comprehensive capacity vector of player v_i
$Score(v_i), Level(T)$	The comprehensive score of player v_i and team T
$Goal(T)$	The number of goals scored by team T in a match
$Match_i, Opponent_i$	The Huskies' ith match and opponent
θ	The parameter (column) vector
$\{S, A, P_{Sa}, \gamma, R\}$	The Markov 5-tuple set

4 Models

4.1 Football Passing Network

Players on the field can be taken as nodes and ball passings between players constitute links between nodes. Connections between some players may become stronger if they pass the football more frequently with one another. To better display such connections, we finally construct a directed network, which can be seen as a weighted directed graph.

We use graph theory to demonstrate this network. Take the network as graph $G(V, E)$, where V is the set of nodes, E is the set of weighted edges. The weight between the ith node to the jth node, w_{ij}, is the number of passing times from v_i to v_j. By analysing specific characteristics of $G(V, E)$, we are able to identify network motifs such as dyadic and triadic configurations. Moreover, we can gain insights into the deep nature of the network.

An example of passing network is shown in figure 2, in which bench players are also counted in. It is the 6th match and Huskies won the game 2-1. The network illustrates clearly that Huskies performed well and had a overall position forward, which means a more aggressive macro network pattern.

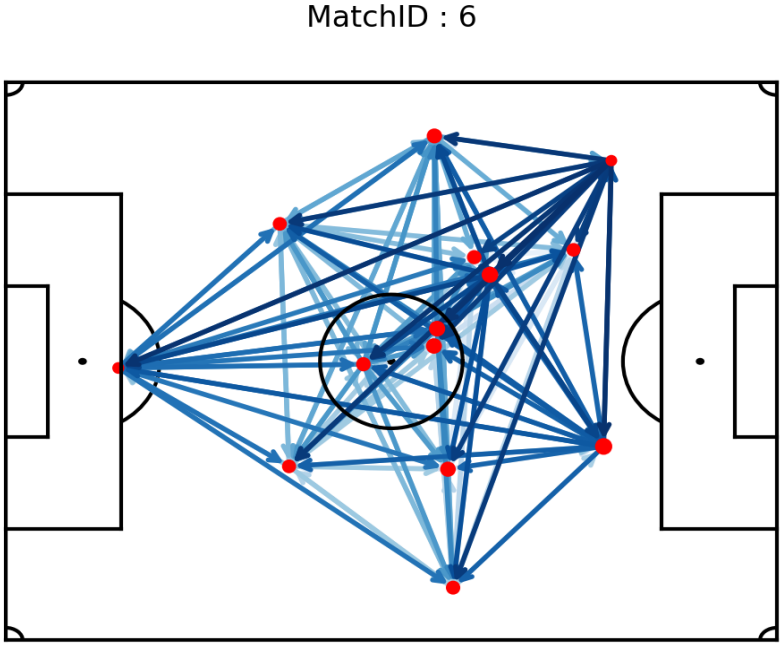


Figure 2: An example of passing network

4.1.1 Average Path Length

For the weighted directed graph $G(V, E)$, the topology distance from node v_i to a neighboring node v_j is defined as

$$l_{ij} = \frac{1}{w_{ij}}.$$

Besides, the distance from node v_i to node v_j, d_{ij}, is defined as the sum of topology distance on the shortest path. Here "shortest" means the sum of topology distance is the smallest. Then the average path length of $G(V, E)$ is

$$L = \frac{2}{N(N-1)} \sum_{1 \le i \le j \le N} d_{ij}.$$

The average path length is an macro indicator to measure the closeness of cooperation between players. The smaller the average path length of a team is, the closer their cooperation between players is.

To calculate the short path length d_{ij}, we introduce the Dijkasta Algorithm and Laplace Smoothing Method (LSM). The Python code can be found in the Appendix. We finally get the average short path length for Huskies in 38 matches, which is 1.071876. To better understand that digit, we compare it with Opponent6, whose average short path length in Match 6 and Match 21 against Huskies is 1.135139. It can be seen that Huskies and Opponent6 have similar network connectivity, but the latter is a bit higher, which means Opponent6 might have a tighter teamwork in the two matches.

4.1.2 Clustering Coefficient

To gain insight into the network patterns in micro aspects, we generalize the definition of clustering coefficient in [2]. We consider different network motifs and define specific clustering coefficient as a measurement.

We firstly normalize the weight as following:

$$\hat{w}_{ij} = \frac{w_{ij}}{\max\limits_{j,k} w_{jk}} \in [0, 1].$$

For two-player cooperation, we define the Second-order Clustering Coeffi-

cient for node v_i as:

$$C_i^{(2)} = \frac{\sum_{j \neq i} \frac{1}{\hat{w}_{ji}} \cdot \hat{w}_{ij}}{\sum_{j \neq i} \hat{w}_{ij}}$$

and the Second-order Clustering Coefficient for the network as

$$C^{(2)} = \frac{1}{N} \sum_{i=1}^{N} C_i^{(2)},$$

where N is the number of nodes in $G(V, E)$.

The formula can be easily understood by taking $\frac{1}{\hat{w}_{ji}}$ as an enhancement of $\hat{w}_{ij}$: if player i and player j perform two-player cooperation, which means after catching the ball from player i, player j will pass it back, then connection between v_i and v_j should be enhanced. Therefore $C_i^{(2)}$ is a good indicator for dyadic configurations.

Similarly, for threesomes, we define the Third-order Clustering Coefficient for node v_i with v_j and v_k as:

$$C_i^{(3)} = \frac{\sum \frac{1}{\hat{w}_{jk}} \hat{w}_{ij} \hat{w}_{ik}}{\sum \hat{w}_{ij} \hat{w}_{ik}},$$

where i, j, k in the equation should have different values. For ball passings from player i to player j and k, if player j and k have ball connections, the network will be enhanced.

The overall Third-order Clustering Coefficient for the network is the mean of all personal clustering coefficients:

$$C^{(3)} = \frac{1}{N} \sum_{i=1}^{N} C_i^{(3)},$$

which is a good measurement for triadic configurations.

Note that clustering coefficient is always larger than 1, we need to consider it relatively. For example, we calculate the clustering coefficient in the 6th match, finding that Huskies has 9.410498 and 6.514914 of second and third order clustering coefficient, while Opponent6 has 6.523558 and 4.625988 respectively. We can concluded that in the 6th match, Huskies has more dyadic as well as triadic configurations than Opponent6, which fits well with the actual result. We also calculate the 21th match with the same opponent, but this time Huskies lost the

game since it has smaller $C^{(2)}$ and $C^{(3)}$, 5.005333 and 3.726794, while Opponent6 gets 6.656953 and 4.489826 respectively.

4.2 Player Analysis Model

To analyse a team's level, it is important that we give a comprehensive ability index for the teammates. In order to better understand a player, we define five aspects that make up the player's total ability: *attack, speed, defense, teamwork and technique* respectively. Each aspect is determined by the player's performance in a match. For example, if the player made a shot, we add some points in his attack ability. If the player is a goalkeeper and he made a save attempt, we think this reflects his bravery and defense ability, then we add some points in it. Concrete contents are shown in Figure 3.

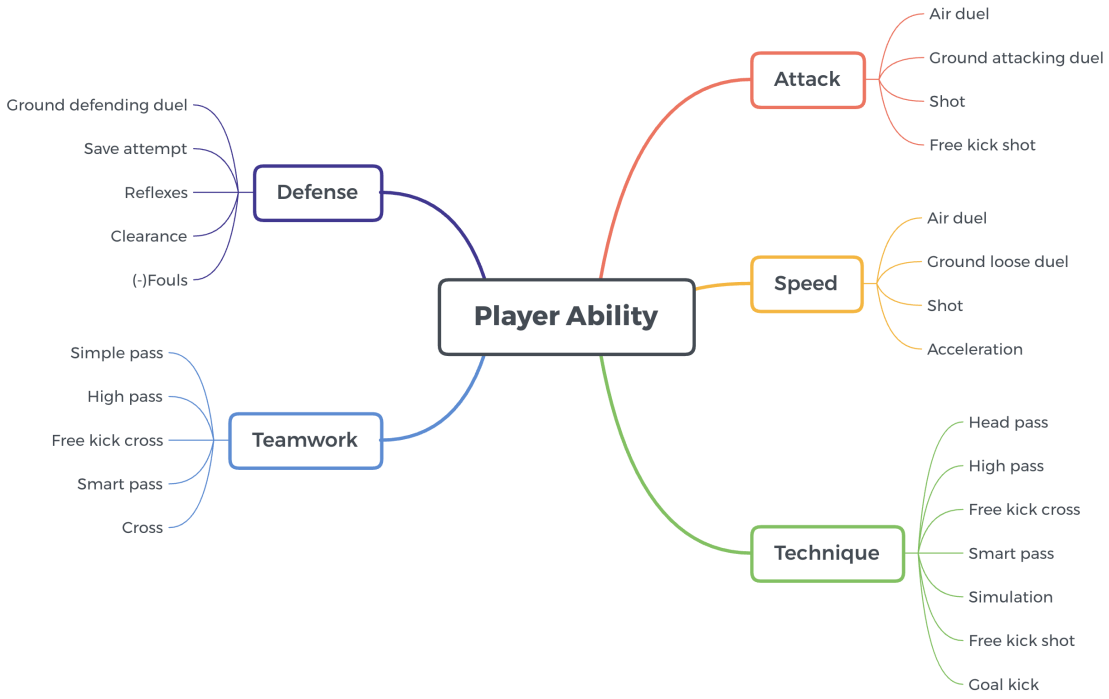


Figure 3: Standards to score every player

4.2.1 Radar Graph

For a specific player, by considering his performance in a match, we can get a radar graph that illustrates his ability displayed in the match. We get the five

ability values by auto-updating. We construct an iterative formula

$$x_n = \frac{(para - 1)x_{n-1} + 1}{para + 1 - x_{n-1}},$$

which let the ability value auto-update in a suitable speed, and finally converge to 1.0. *para* is the parameter and by changing the value of *para*, we can control the iteration speed. Relative mathematical proofs can be found in the appendix.

We get Huskies' each player' radar vector in every match. As shown in Figure 4, we take advantage of these vectors and by averaging method, and get the team's radar graph which roughly reflects the teammates' performance in five aspects during the whole season of 38 matches.

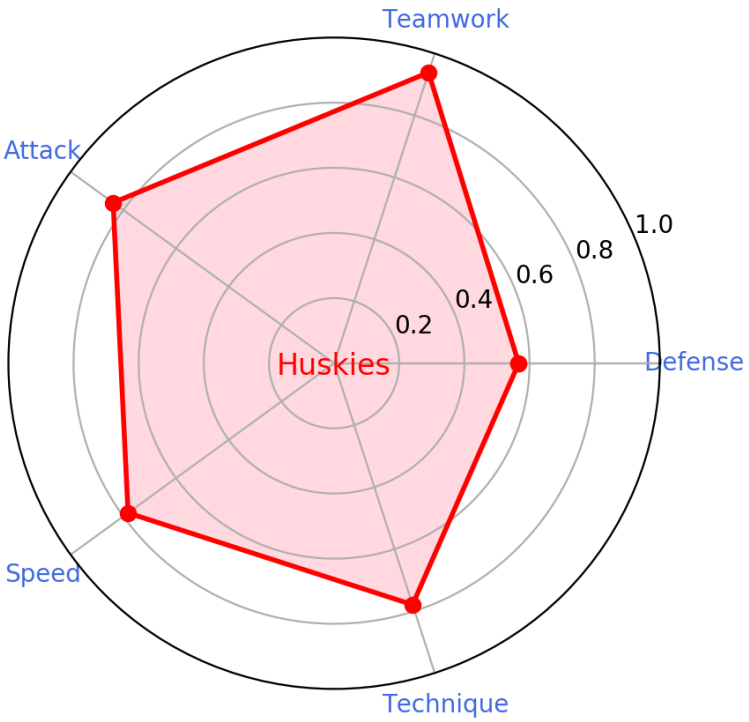


Figure 4: Huskies' radar map

4.2.2 The Scoring of Each Player

In this model, we are going to score the performance of each player, and then judge the playing level of a team. If a team scores higher in playing level than its opponent, it's more likely to score goals and less likely to receive goals.

Given the comprehensive capacity vector $R(v_i) = [R_1(v_i), R_2(v_i), \cdots, R_n(v_i)]^T$ of the ith team member, his score is a linear combination of his comprehensive capacity index. Let $\theta \in \mathcal{R}^{n+1}$ be the parameter vector, i.e. for each player v_i, $Score(v_i) = \theta^T \begin{bmatrix} R(v_i) \\ 1 \end{bmatrix}$, and, therefore, for each team T, the performance level of a team is the mean of its team members' scores

$$Level(T) = \frac{1}{|T|} \sum_{v_i \in T} Score(v_i) = \frac{1}{|T|} \sum_{v_i \in T} \theta^T \begin{bmatrix} R(v_i) \\ 1 \end{bmatrix}$$

According to our previous assumptions, we can assume that the goals and losses of T_a against his opponent T_b subject to Poisson distribution $Poisson\left(\alpha \frac{Level(T_a)}{Level(T_b)}\right)$ and $Poisson\left(\alpha \frac{Level(T_b)}{Level(T_a)}\right)$ respectively. For the sake of convenience, we assume that the coefficient $\alpha = 1$. Let $Level(T_a)/Level(T_b) = t$, $Level(T_b)/Level(T_a) = t^{-1}$, so

$$Goal(T_a) \sim Poisson\left(\alpha \frac{Level(T_a)}{Level(T_b)}\right) = Poisson\left(\frac{Level(T_a)}{Level(T_b)}\right) = Poisson(t)$$

$$Goal(T_b) \sim Poisson\left(\alpha \frac{Level(T_b)}{Level(T_a)}\right) = Poisson\left(\frac{Level(T_b)}{Level(T_a)}\right) = Poisson(t^{-1})$$

Then

$$\begin{aligned} P(Match_i) &= P(Goal_i(Huskies))P(Goal_i(Opponent_i)) \\ &= \frac{t^{Goal_i(Huskies)}}{Goal_i(Huskies)!} e^{-t} \frac{t^{-Goal_i(Opponent_i)}}{Goal_i(Opponent_i)!} e^{-t^{-1}} \\ &= e^{-t-t^{-1}} \frac{t^{Goal_i(Huskies)}}{Goal_i(Huskies)!} \frac{t^{-Goal_i(Opponent_i)}}{Goal_i(Opponent_i)!} \end{aligned}$$

Ultimately, obtain θ by maximizing the log likelihood function:

$$\theta = \underset{\theta}{argmax}\,[l(\theta)] = \underset{\theta}{argmax} \left[\sum_{i=1}^{m} \ln P(Match_i) \right],$$

$$\text{s.t. } [1,1,1,1,1,1]\theta = 1.$$

This final step is completed by gradient descent by programs. According to the data set, we finally fix the parameter vector

$$\theta = [0.16, 0.31, 0.20, 0.09, 0.14, 0.10]^T$$

So the score of each player v_i is

$$Score(v_i) = \theta^T \begin{bmatrix} R(v_i) \\ 1 \end{bmatrix} = [0.16, 0.31, 0.20, 0.09, 0.14, 0.10] \begin{bmatrix} R(v_i) \\ 1 \end{bmatrix}$$

4.2.3 The Scoring of the Whole Team

After computing θ, we can use it to rectify and rescale our radar model, and thus, the scoring for the whole team becomes practicable.

We have already worked out the capability vector for each player in each match. Then, all of these vectors are placed together into a matrix M_{m*6} (every element in the sixth column is 1) . So we have

$$Score(Huskies) = \left[\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m}\right] M\theta = 0.828879$$

as our final overall measurement on this team. It must be noted that this measurement can also be applied to other teams in the league in exactly the same way, so long as we have enough game records. If we got our opponents' information and understood the balance of the levels of both sides, it would definitely help to decide whether to take counter strategies. I.e. in most of cases, it's wise to follow established strategies against weaker opponents and take counter strategies against stronger opponents.

4.3 Tactics for Attacks and Defences

4.3.1 Decisions for Attacks

In this model, we take the network feature and personal scores as our inputs and make decisions for attacks using Markov Decision Process (MDP). We first divide the court into 13 regions as shown in Figure 5, each numbered 1-5 corresponding to its distance to opponent's gate. And we also assume that a player can only move forward when attacking, i.e. moving from one region to another region that is closer to opponent's gate.

Let's consider the Markov 5-tuple set:

$$\{S, A, P_{Sa}, \gamma, R\}.$$

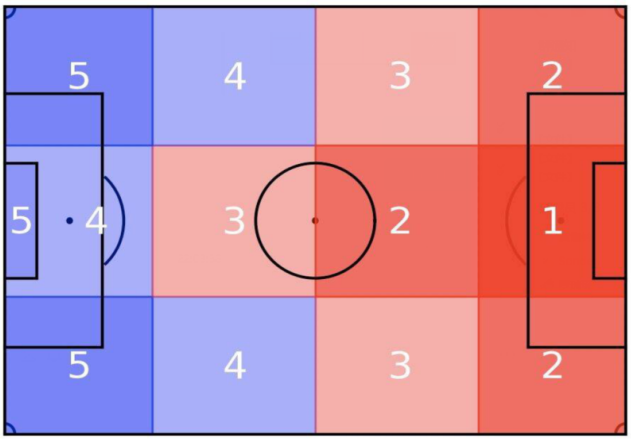


Figure 5: Divided Football Field

Define our set of states

$$S = \{r, PLY_1, PLY_2, \cdots, PLY_{11}, g, b\},$$

where

$$r = \begin{cases} 0, & \text{if the model ceases} \\ 1, & \text{if the model is running} \end{cases}$$

$$g = \begin{cases} 0, & \text{if haven't scored this one} \\ 1, & \text{if succeeded to score this one} \end{cases}$$

$$b = \begin{cases} 0, & \text{if the ball is controlled by the opponent} \\ i, & \text{if the ball is controlled by our } i\text{th player}, i = 1, 2, \cdots, 11 \end{cases}$$

PLY_i is the region the ith player stands in.

Define set of actions for the player controlling the ball

$$A = \{dribble, pass, shot\}$$

state transition distributions (the definition of s_{hope} and s_{fail} will be given later)

$$P_{s,a}(s' = s_{hope}) + P_{s,a}(s' = s_{fail}) = 1$$

discount factor

$$\gamma = 0.98$$

and reward function R (due to the fact that the more forward defenders rush, the

more risky it is when losing control of the football)

$$R(g=0,b=0)=\frac{-20}{P_b\sum_{i=8}^{11}PLY_i}$$

$$R(g=1)=40$$

At initial state s_0, we choose action a_0, get to $s_1 \sim P_{s_0a_0}$, then choose a_1, and get to $s_2 \sim P_{s_1a_1}\cdots$ By taking actions $\{a_0, a_1, \cdots, a_n, \cdots\}$ overtime, we get the sequence of states $\{s_0, s_1, \cdots, s_n, \cdots\}$ and our total payoff

$$Payoff = R(s_0) + \gamma R(s_1) + \cdots + \gamma^n R(s_n) + \cdots$$

So we'd like to choose consequent actions, or namely our policies, to maximize the $E[Payoff]$.

More concretely, we define V^π, V^* and π^* as following:

(1) For any policy π, define value function:

$$V^\pi(s) = E\left[R(s_0) + \gamma R(s_1) + \cdots + \gamma^n R(s_n) + \cdots \mid \pi; s_0 = s\right]$$

(2) We can derive the Bellman Equation:

$$V^\pi(s) = R(s) + \gamma \sum_{s'} P_{s\pi(s)}(s')V^\pi(s')$$

(3) Define the optimal value function:

$$V^*(s) = \max_\pi [V^\pi(s)]$$

(4) Define the optimal policy:

$$\pi^* = \underset{\{a_n\}}{argmax}\left[\sum_{s'} P_{sa}(s')V^*(s')\right]$$

Back to our next-state distribution, we assume that:

(1) Forwards and midfielders (player 1-7) can reach any region on the court, while defenders' (player 8-10) farthest reach is region 2 and the goalkeeper's

(player 11) is region 4;

(2) For the next state, all players move to neighbouring region with a smaller number if within his reach.

(3) No matter what action the player takes, the value of b of the next state (namely b') is himself or the teammate he passes to if he does the job successfully (s_{hope}), and is 0 if he fails to do that (s_{fail}); g turns to 1 (s_{hope}) if and only if the player decides to shot and the ball goes in; after b turns to 0 or g turns to 1, run turns to 0 such that S ceases at current state.

Therefore, according to our assumptions, let

$$P_{s,dribble}(s_{hope}) = \alpha\beta Score(b)\frac{1}{PLY_b}$$

$$P_{s,dribble}(s_{fail}) = 1 - \alpha\beta Score(b)\frac{1}{PLY_b}$$

$$P_{s,pass}(s_{hope} \mid b') = \beta Score(b)\frac{w_{bb'}}{\sum_{p\neq b} w_{bp}}$$

$$P_{s,pass}(s_{fail}) = 1 - \beta Score(b)$$

$$P_{s,shot}(s_{hope}) = \alpha Score(b)\frac{1}{PLY_b^2}\sqrt{\frac{1}{11}\sum_{1\le i\le 11}\frac{1}{PLY_i^2}}$$

$$P_{s,shot}(s_{fail}) = 1 - \alpha Score(b)\frac{1}{PLY_b^2}\sqrt{\frac{1}{11}\sum_{1\le i\le 11}\frac{1}{PLY_i^2}}$$

where

$$\alpha = \begin{cases} 0.8, & \text{if } PLY_b = 1 \\ 1, & \text{if } PLY_b > 1 \end{cases}$$

$$\beta = \frac{1}{5}\sum_{8\le i\le 11}\frac{PLY_i}{5}$$

Set

$$s_0 = \{1, 2, 3, 3, 3, 3, 3, 3, 4, 3, 4, 4, 0, 2\}$$

$$s_0 = \{1, 2, 3, 3, 3, 3, 3, 3, 4, 3, 4, 4, 0, 3\}$$

$$s_0 = \{1, 2, 3, 3, 3, 3, 3, 3, 4, 3, 4, 4, 0, 1\}$$

$$s_0 = \{1, 3, 4, 4, 5, 4, 4, 5, 5, 5, 5, 5, 0, 9\}$$

as initial state for side-attack(left), side-attack(right), middle-attack, and counter-attack, respectively.

Finally, by applying value iteration, the optimal policy π^* for each state is found. This step is completed by computer programs shown in appendix.

The general ideas we learnt from optimal policy network is that:

(1) Among these attacking ways, counterattack is more threatening and less risky for this team in general, and side attacks are usually more threatening than middle attacks;

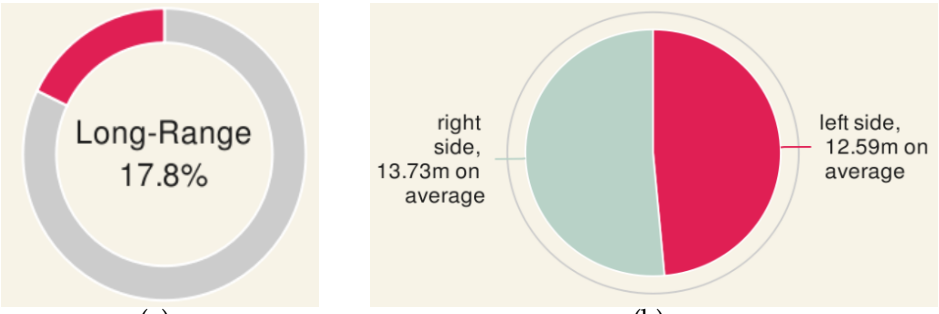
(2) Attacks launched from the left side have a better effect over those launched from the right side;

(3) For midfielders and defenders, they are advised to use a combination of passing and dribbling before entering threatening area, and then complete 1-3 passing before ultimately finding a threatening position for his teammates or himself to shoot;

(4) As to forwards, they should be brave to shoot in both region 1 and 2.

4.3.2 Decisions for Defences

We make decisions for defences in a more vivid and direct way based on the data extracted from "fullevents.csv", which is depicted in Figure 6 :



(a) (b)

Figure 6: Data Extracted

(1) 17.82% of the shot attempts of his opponents are long-ranges ($\geq 20m$), and this is considered a relatively small amount. This suggests that Huskies tends to let the opponents enter some risky areas to take a shot, which is a signal that Huskies should put more emphasis on stopping opponents' oppressive attacks.

(2) Among 422 shot attempts of his opponents, 48.58% comes from their left sides, while 51.42% comes from right sides, which are almost the same. But the real point is that those comes from the left shoot at 12.59m, 1.14m closer to our gate than that from another side. This is partly due to poor cooperation between left-defend(LD) and middle-defend(MD) or due to the limited level of LD, and therefore, buying players for this position or strengthening trainings on this side may be a good choice.

5 Optimization and Extension for Our Model

5.1 Opinions in Group Dynamics

Some opinions in group dynamics may be helpful here in the extending of our model [7].

According to Steiner's theory(1972), a group's actual productivity is equal to its potential productivity minus process losses [8]:

$$ActualProductivity = PotentialProductivity - ProcessLosses.$$

And loses are normally due to two main areas:

- **Motivational faults:** some team members are not giving 100%.
- **Coordination faults or losses:** team interacts poorly or have ineffective strategies.

Thus, one can predict that team A will perform better than team B if any of the following scenarios is present:

- Team A possesses greater relevant resources than team B does and experiences equal process losses, or
- Team A possesses equal relevant resources but experiences fewer process losses than team B does, or
- Team A possesses greater relevant resources and experiences fewer process losses than team B does.

Accordingly, the primary role of a coach is to increase relevant resources through instruction, training and recruiting and reduce process losses through strategies for motivating players and combining their contributions.

Besides, theory of group dynamics also give some advise for both team members and the coach. To help build an effective team climate, team members should be responsible for their activities, resolve conflict quickly, get to know each other and help each other both on and off the field; the coach should get to know their athletes, communicate effectively, ensure everybody knows their role and keep changes(except the diversity of strategies) in the team to a minimum.

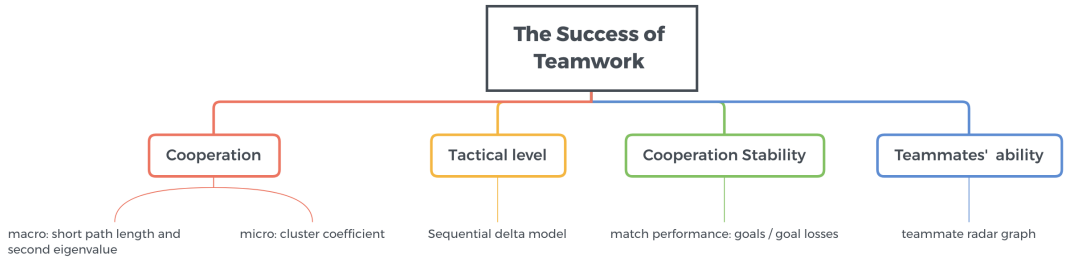


Figure 7: The Success of Teamwork

5.2 Discussion on Quaternary Configuration

We have discussed dyadic and triadic configurations in the above. Here we do some discussion on quaternary configuration.

Denote four players are A, B, C, D respectively, and A is now controlling the ball, D is near opponent's gate. Theoretically, passing between four players is more flexible and hard to be predicted by opponents, but the fact is not that ideal.

- If they stands in lie, it's no matter how many players are in cooperation, because there is only one path.
- If they stands at four vertices of quadrilateral:
  - If the quadrilateral is a square or rectangle, $\overline{AD}$ is the diagonal line, then the passing from A to D is too long to realize. A has to pass the ball to B or C, since they won't pass it back to A because of the long distance between A and D, then the cooperation is no different than the triadic one.

- If the quadrilateral is convex and no one has the same x-coordination, then A has to pass his ball to the player who is accessible, then, as discussed before, the cooperation is no different than the triadic one.
- If the quadrilateral is not convex, then the question is simple: just pass it to the one who is surrounded by other three players.

As discussed above, quaternary configuration is no better than triadic or dyadic one, besides triangle is more stable and two fixed players are more tacit, so we recommend dyadic and triadic configurations.

6 Sensitivity Analysis

6.1 Stability of Clustering Coefficient

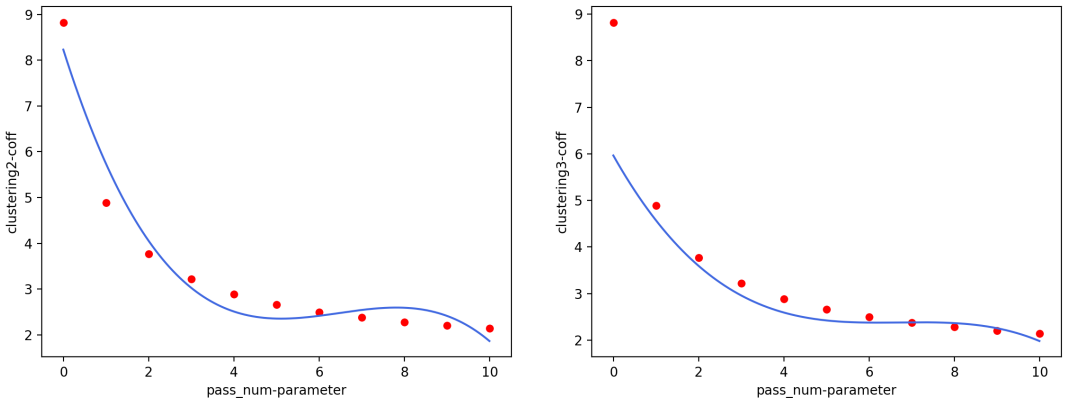
Though clustering coefficient is an excellent indicator, there's still a little problem. When we normalize the weights, the maximum of adjacent matrix is divided. However, if we only change the maximum element of adjacent matrix for a bigger number, fixing others elements, the fluctuation of clustering coefficient may be very big. Back to the definition of adjacent matrix, we know that the change of the maximum element only means one player passes or receives from an other player, and such change doesn't influence the pattern of passing network, which is not in accordance with clustering coefficient.

As is shown in Figure 8, when the maximum becomes bigger, clustering coefficient varies a lot in the beginning.

6.2 Stability of Average Path Length

When we use Laplace Smoothing Method to make data easier to analysis, we put a Laplace parameter α in. We change the value of α and observe the amount of change for the average path length of the network.

From Figure 9 next page, we can conclude that when we add suitable parameter α, the short path length has good stability. In addition, we found that when LSM is not used, which means the Laplace parameter is equal or very close to 0, the short path length can vary a lot. Hence, it is important to use LSM to make the model stabler.



(a) second-order clustering (b) third-order clustering

Figure 8: Sensitivity analysis of clustering coefficient

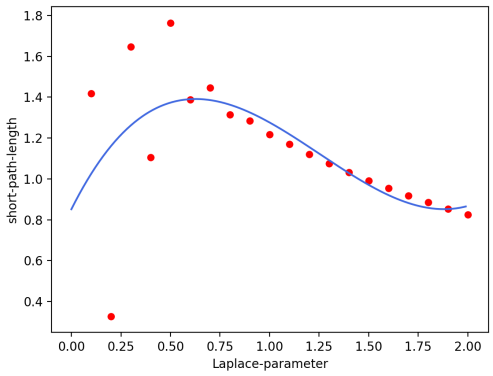


Figure 9: Sensitivity analysis of average path length

7 Strengths and Weaknesses

7.1 Strengths

- Almost every aspect of the team's performance can be quantified such that the old judging system (consisting of traditional indicators such as shot, goal, assistance, mistake $\cdots$) is improved.
- Personal performance and the team's performance are related, and in this process cooperation is counted in.
- Adopt different strategies for teams with different styles so that our model can be extended.

- By applying Laplace Smoothing Method, outcome becomes much more stable and close to reality.

7.2 Weaknesses

- The stability of adjacent matrix is not so good. For example, if a player passes the ball too many times(much more than others), maybe he is the core of the team. But the indicator clustering coefficient's change will be very big, leading us to misjudge the team's style.
- Substitution of players is not considered fully in our model.

8 Conclusions

In general, we develop three significant models in our paper: the passing network model, the scoring model for persons and teams and the decision model for attacks and defences. Here are the main ideas concluded from our models and analyses:

(1) We construct ball passing network for players and use graph theory to tap into its nature. We define topology short path length as a macro indicator of the team's cooperation and generalize the clustering coefficient for the measurement of the network patterns. We calculate out these features and the data can be considered when look into the cooperation level of the team.

(2) We draw radar map for each player and score the general performance for the whole team, both of which can be used as the reference for next season's strategic bets.

(3) Markov Decision Process shows the optimal policies for attacking organization. For Huskies, in general, i.e. counterattacks are more threatening and less risky, left side is more powerful than the right one, defenders may also participate in some attacks, long-range shot should also be a decent choice.

Finally, based on these three models and some basic theories in group dynamics, we attempt to extract a universal measurement standards, concerning cooperation, tactical level and teammates' ability.

References

[1] Xiaofan Wang, Xiang Li, Guanrong Chen. Network Science: An Introduction, 2012. *Higher Education Press*.

[2] Buldú, J.M., Busquets, J., Echegoyen, I. *et al*. Defining a historic football team: Using Network Science to analyze Guardiola's F.C. Barcelona, 2019. *Sci Rep*, 9, 13602.

[3] Pappalardo, L., Cintia, P., Rossi, A. *et al*. A public data set of spatio-temporal match events in soccer competitions, 2019. *Sci Data*, 6, 236.

[4] Cintia, P., Giannotti, F., Pappalardo, L., Pedreschi, D., Malvaldi, M. The harsh rule of the goals: Data-driven performance indicators for football teams, 2015. *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 1-10, 7344823.

[5] Duch J., Waitzman J.S., Amaral L.A.N. Quantifying the performance of individual players in a team activity, 2010. *PLoS ONE*, 5: e10937.

[6] GÜRSAKAL, N., YILMAZ, F., ÇOBANOĞLU, H., ÇAĞLIYOR, S. Network Motifs in Football, 2018. *Turkish Journal of Sport and Exercise*, 20(3), 263-272.

[7] Eys, M., & Kim, J. Team building and group cohesion in the context of sport and performance psychology, 2017. *Oxford Research Encyclopedia of Psychology*.

[8] Thelma S. Horn. Advances in Sport Psychology(3rd edition), 2012. *Human Kinetics*.

[9] http://ee.usc.edu/stochastic-nets/

[10] https://see.stanford.edu/Course/CS229

[11] https://footballdata.wyscout.com/

Appendices

Appendix A Markov Decision Process

We hereby only show the sketch of our main function in the implement of Markov Decision Process and neglect those subroutines due to limited space.

Language: C++

```
int main(int argc, char* argv[])
{
    initWeight(); //the passing weight for players
    initPolicy(); //the policies players take at each state
    setState(); //set all states
    initReward(); //set reward functions
    int cnt = 2*n;
    while (cnt--) //iterates 2*n times
    {
        for (int i=0;i<n;i++) //go over all states
        {
            int v0, v1, v2, v;
            //try all three actions
            updatePolicy(i, 0); v0 = tryAction(i, 0);
            updatePolicy(i, 1); v1 = tryAction(i, 1);
            updatePolicy(i, 2); v2 = tryAction(i, 2);
            v = findmax(i, v0, v1, v2);
            //find optimal value & return the index
            updatePolicy(i, v);
            //fix optimal policy network (greedy)
            updateValue(i); //update value network
        }
    }
    for (int i=1;i<n;i++)
        printPolicy(i); //print the outcomes
    return 0;
}
```

Appendix B Calculating the Clustering Coefficient

Here is the code for the calculation of clustering coefficients.

Language: Python

```
  import   numpy as np
import matplotlib.pyplot as plt

A = np.load('./Adjacent Matrix/Huskies_A.npy', allow_pickle=True)
M21 = np.load(
    './Adjacent Matrix/Match21Opponent6_A.npy', allow_pickle=True)
M6 = np.load(
    './Adjacent Matrix/Match6Opponent6_A.npy', allow_pickle=True)

def get_cluster2_coff(a):
    m = len(a)
    w = np.zeros((m, m))
    a += 1
    amax = np.amax(a)
    for i in range(m):
        for j in range(m):
            w[i][j] = a[i][j] / amax
    coffs = np.zeros(m)
    for i in range(m):
        m1 = 0
        m2 = 0.0
        for j in range(m):
            if i != j:
                m1 += 1 / w[j][i] * w[i][j]
                m2 += w[i][j]
        if m2 == 0:
            continue
        coffs[i] = m1 / m2

    return np.mean(coffs)

def get_cluster3_coff(a):
    m = len(a)
    a += 1
    amax = np.amax(a)
    w = np.zeros((m, m))
    for i in range(m):
        for j in range(m):
            w[i][j] = a[i][j] / amax

    coffs = np.zeros(m)
    for i in range(m):
        m1 = 0.0
        m2 = 0.0
```

```
        for j in range(m):
            for k in range(m):
                if i != j and i != k and j != k:
                    m1 += 1 / w[j][k] * w[i][j] * w[i][k]
                    m2 += w[i][j] * w[i][k]

        if m2 == 0:
            continue
        coffs[i] = m1 / m2

    return np.mean(coffs)

def get_cluster2s():
    n = len(A)
    coffs = []
    for i in range(n):
        coffs.append(get_cluster2_coff(A[i]))
    C = np.mean(coffs)

    coff6 = coffs[5]
    coff21 = coffs[20]
    M6_opponent6_coff = get_cluster2_coff(M6)
    M21_opponent6_coff = get_cluster2_coff(M21)

    print("In the 6th match, Huskies' cluster2 coefficient    ", coff6)
    print("In the 6th match, Opponent6's cluster2 coefficient",
              M6_opponent6_coff)
    print("In the 21th match, Huskies' cluster2 coefficient    ",
              coff21)
    print("In the 21th match, Opponent6's cluster2 coefficient",
              M21_opponent6_coff)
    print("Huskies' mean cluster2 coefficient in 2 matches with
              Opponent6: ", (coff6 + coff21) / 2)
    print("Opponent6's mean cluster2 coefficient : ",
              (M6_opponent6_coff + M21_opponent6_coff) / 2)
    print("Huskies' mean cluster2 coefficient in 38 matches : ", C)

def get_cluster3s():
    n = len(A)
    coffs = []
    for i in range(n):
        coffs.append(get_cluster3_coff(A[i]))
    C = np.mean(coffs)

    coff6 = coffs[5]
```

```
    coff21 = coffs[20]
    M6_opponent6_coff = get_cluster3_coff(M6)
    M21_opponent6_coff = get_cluster3_coff(M21)

    print("In the 6th match, Huskies' cluster3 coefficient    ",
              coff6)
    print("In the 6th match, Opponent6's cluster3 coefficient",
              M6_opponent6_coff)
    print("In the 21th match, Huskies' cluster3 coefficient    ",
              coff21)
    print("In the 21th match, Opponent6's cluster3 coefficient",
              M21_opponent6_coff)
    print("Huskies' mean cluster3 coefficient in 2 matches with
              Opponent6: ", (coff6 + coff21) / 2)
    print("Opponent6's mean cluster3 coefficient : ",
          (M6_opponent6_coff + M21_opponent6_coff) / 2)
    print("Huskies' mean cluster3 coefficient in 38 matches : ", C)

get_cluster2s()
print()
get_cluster3s()
```

Appendix C Calculating the Average Path Length

Here is the code for the calculation of the average path length using Dijkstra Algorithm.

Language: Python

```
import numpy as np
# load data
A = np.load('./Adjacent Matrix/Huskies_A.npy', allow_pickle=True)
M21 = np.load(
    './Adjacent Matrix/Match21_Opponent6_A.npy', allow_pickle=True)
M6 = np.load(
    './Adjacent Matrix/Match6_Opponent6_A.npy', allow_pickle=True)

# dijkasta algorithm

def shortPath_Dijkstra(graph, v0):
    k = 0
    n = len(graph)
    final, P, D = [0] * n, [0] * n, [0] * n
```

```
    for i in range(n):
        D[i] = graph[v0][i]
    D[v0] = 0
    final[v0] = 1
    for v in range(1, n):
        min = float("Inf")
        for w in range(0, n):
            if not final[w] and D[w] < min:
                k = w
                min = D[w]

        final[k] = 1
        for w in range(0, n):
            if not final[w] and min + graph[k][w] < D[w]:
                D[w] = min + graph[k][w]
                P[w] = k
    return D

# Transform the distance into topology distance
# Use Laplace Smoothing Method with parameter 0.5
def trans_dist(m):
    n = len(m)
    for i in range(n):
        for j in range(n):
            m[i][j] = 1 / (m[i][j] + 0.5)

# Get short path length
def get_pathlen(a):
    trans_dist(a)
    n = len(a)
    d = 0
    short_path = []

    for i in range(n):
        tmp = np.array(shortPath_Dijkstra(a, i))
        d += np.sum(tmp)
    if n == 1:
        return 0

    return 2 * d / (n * (n - 1))

# Get mean short path length in the whole season of 38 matches
def get_average_path_len():
    ans = []
    for i in range(38):
```

```
        tmp = 'The ' + str(i + 1) + 'th ' + \
            "match's average path length for Huskies : "
        tmp2 = round(get_pathlen(A[i]), 6)
        print(tmp, tmp2)
        ans.append(tmp2)
    return ans

print('mean of 38 : ', round(np.mean(get_average_path_len()), 6))
m6_path = get_pathlen(M6)
m21_path = get_pathlen(M21)
print(round(m6_path, 6))
print(round(m21_path, 6))
print(round((m6_path + m21_path) / 2, 6))
```

Appendix D Proof of Iteration

When computing personal score, we want to ensure that it converges to 1, so we design the iteration from the function $a_n = 1 - \frac{c}{n}$. We have

$$\frac{c}{n} = 1 - a_n$$

$$\frac{c}{n+1} = 1 - a_{n+1}$$

Put these two formulas together, we can derive that:

$$a_{n+1} = \frac{(c-1)a_n + 1}{c + 1 - a_n}$$

Therefore, we choose this function as our iteration tools in computing, where the constant c can be modified to adapt the convergence speed.