University of Colorado - Boulder

APPM 3310: Matrix Methods

Final Project Report

Ranking of Sports Teams

Author:
Nick Larson[a]

[a]109409056 || Section: 003

Professor: Gregory Beylkin
Professor: Ian Grooms

28 April, 2021

Abstract

The objective of this report is to use three different mathematical methods to rank sports teams. The results of these methods will be compared to the ranking of sports teams based off of their record, and they will also be compared against each other. This will be done by applying the methods to the 2020 NFL regular season, and using the results to predict the outcome of the playoffs. This paper did indeed show that a teams record alone is not the best indication of how good that team is. It also showed the power of the methods analyzed, which were founded using linear algebra. The rankings that these methods produced were found to be effective when used to predict the outcome of the 2020 NFL playoffs, which validated their accuracy.

I. Background

The ranking of sports teams is important for many reasons. In most sports, teams earn a chance to compete for a championship based off of their ranking. For example, in NCAA football, a team must earn a spot in a bowl game based on their ranking alone. The four teams with the best rankings earn a spot in the top two bowl games, and the winner of these bowl games face each other in the NCAA Championship. In college basketball, March Madness is a single game elimination tournament in which teams are matched up in a bracket format, and the winner of the tournament is crowned as NCAA Champions. The bracket for this tournament is formed so that the highest ranked teams have the easiest schedule in the tournament. For example, the number one seeds on each side of the bracket (the highest ranked teams) play the sixteenth seeds in the first round of the tournament (the worst ranked teams). The first seeds then go on to play the winner of the game between the eighth and ninth seeds. Put simply, the better a teams ranking, the easier their games will be, and their chances of winning the tournament are higher. Ranking sports teams could also be useful to predict the outcome of a game, which could be of interest to the average person when it comes to formulating a March Madness bracket, or sports betting. This project will investigate the use of linear algebra to rank sports teams. The investigated methods aim to be far more accurate than ranking teams solely based on their records for many reasons. Some of which are listed below:

- Not every team will play each other, therefore some teams will get a very easy schedule and some teams will get a challenging schedule.
- Ranking teams off their records doesn't take into account point differentials. For example, two teams with similar records and similar schedules should be ranked differently if one team won their games by significantly more and lost their games by significantly less.
- Circumstances such as the whether, home field advantage, or injuries are also not taken into account by only looking at a teams record. If the majority of a teams starters are injured, or if a football team who relies heavily on passing the ball plays a game in bad conditions, that is not a true reflection of the teams skills in more ideal conditions.

At the most basic level, a linear system can be constructed in the form $A\vec{r} = \vec{b}$ where A is an mxn matrix and $\vec{r}$ and $\vec{b}$ are vectors. The number of rows of the matrix, m, is the total number of games played. The number of columns, n represents the number of teams that are being considered. The matrix A is determined by filling each element with a 1 representing a win, a -1 representing a loss, and a 0 representing a draw or bye. The vector $\vec{r}$ is the ranking vector which contains elements that correspond to each team, and the vector $\vec{b}$ has elements that indicate how much the winning team won by. For example, if 3 teams each play each other, and team 1 ends up beating team 2 by 14 points and losing to team 3 by 6 points, and team 2 beats team 3 by 3 points, the linear system that represents this outcome will be:

$$A\vec{r} = \vec{b}$$

$$A = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix} \quad \vec{r} = \begin{bmatrix} team1 \\ team2 \\ team3 \end{bmatrix} \quad \vec{b} = \begin{bmatrix} 14 \\ 3 \\ 6 \end{bmatrix}$$

The solution was obtained using the method of least squares in MATLAB which results in:

$$\vec{r} = \begin{bmatrix} 2.667 \\ -3.667 \\ 1 \end{bmatrix}$$

Based on this, team 1 would be ranked the highest, team 3 would be ranked second, and team 2 would be ranked last. This technique is the most basic technique for the ranking of sports teams, however, problems arise very quickly. This method can only be applied in a scenario where there are three teams, and each team plays the other teams once each. If this method were used to on three teams who played each other more than once, an incomplete system could be formed where a row of zeros in the matrix A would correspond to an actual point differential. Similarly, if more than three teams are considered, then the vector $\vec{b}$ could no longer have its entries correspond to the point differentials between two teams because each row of the matrix A would contain the outcome of more than one game. For these reasons, we must find new methods to investigate which is what the rest of this report will aim to do.

II. Introduction

This report will investigate three methods to more accurately rank sports teams. The first two methods, created by James Keener, assigns a score to each team based on the number of games played, outcome of the games, and rank of the opponent. Both methods produces a matrix A and ranking vector $\vec{r}$, which is also an eigenvector of A. To find the ranking vector, the Perron-Frobenius Theorem is applied which will be covered in a later section. The third method, created by Wes Colley, is based off of win percentage. The linear system created is in the form $C\vec{r} = \vec{b}$. All elements in this system are created based on the number of games a team has won, lost, and played. Similar to the previous method, a ranking vector $\vec{r}$ is produced which allows for comparison between the methods. After all three methods have been discussed, real data from the 2020 NFL regular season will be used to rank all 32 teams using the two different versions of Keener's approach, and Colley's approach.

III. Keener's Approach

A. Version 1

As stated above, Keener's approach involves a matrix A and one of its eignevectors $\vec{r}$ where $A\vec{r} = \lambda\vec{r}$. Before defining A, the preference matrix P must be defined. P is a matrix who's entries must be greater than or equal to zero, and they must be related to the outcome of the game between team i and team j. For simplicity, the matrix P is defined as p_{ij} = the number of times that the ith team has beat the jth team. Next, team one through team i will be given a score, s_i which is defined as follows:

$$s_i = \frac{1}{n_i}\sum_{j=1}^{N} p_{ij}r_j$$

Where n_i is the total number of games that team i plays, N is the total number of teams considered, and r_j is the ranking of team j which must be greater than zero. Conceptually, this means that the score team i receives depends on the number of games team i has played, the outcome of those games, and the strength of there opponent. For the score to be of use, it must be related to to the ranking of the team. In this method, it is assumed that the following is true:

$$\vec{s} = \lambda\vec{r}$$

Since $A\vec{r} = \lambda\vec{r}$ and $\vec{s} = \lambda\vec{r}$, the matrix A must be defined as $a_{ij} = \frac{p_{ij}}{n_i}$. This ensures that $\vec{r}$ is indeed an eignevector of A. To grasp this method more clearly, an example will be done. If there are three teams, and each team has played the other teams twice with the following results:

Teams Playing	Winner
Team 1 vs Team 2	Team 1
Team 1 vs Team 3	Team 3
Team 2 vs Team 3	Team 3
Team 1 vs Team 2	Team 1
Team 1 vs Team 3	Team 1
Team 2 vs Team 3	Team 2

Then the matrix A is as follows:

$$A = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{4} \\ 0 & 0 & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & 0 \end{bmatrix}$$

To find the ranking vector $\vec{r}$, the Perron-Frobenius theorem must be applied. This theorem states that if a square matrix A has all positive entries ($a_{ij} \geq 0$), then it has a unique maximal eigenvalue, and the corresponding eigenvector has positive entries. Since this theroem always allows us to find a positive eigenvector of the matrix A, the ranking vector $\vec{r}$ will be defined such that it is the eigenvector corresponding the the maximal eigenvalue. This process is demonstrated below for the example given above. Using MATLAB, the eignevalues and corresponding eigenvectors of the matrix A were found to be:

$$\lambda_1 = -0.221 + 0.147i \quad \lambda_2 = -0.221 - 0.147i \quad \lambda_3 = 0.442$$

$$\vec{v_1} = \begin{bmatrix} 0.631 \\ -0.253 + 0.467i \\ -0.052 - 0.563i \end{bmatrix} \quad \vec{v_2} = \begin{bmatrix} 0.631 \\ -0.253 - 0.467i \\ -0.052 + 0.563i \end{bmatrix} \quad \vec{v_3} = \begin{bmatrix} 0.724 \\ 0.340 \\ 0.601 \end{bmatrix}$$

The maximum eigenvalue is $\lambda_3 = 0.442$ which means that it's corresponding vector $\vec{v_3}$ is the ranking vector. As previously stated, r_j corresponds to the ranking of the jth team where a higher value correlates to a higher ranking. Therefore, from the example given, team 1 would be ranked the best, team 3 second, and team 2 last. This result makes sense because team 1 won the most games, followed by team 3, and team 2 won the least amount of games.

This method is mathematically sound and does in fact take a few different variables into account, which are listed below:

- The ranking of a team is dependent on the number of games they have played. This ensures that a team won't be ranked high simply because they have a lot of wins due to the high number of games played. If one team were to win 30 of its 60 games, and another team were to win 25 of its 35 games, the team with less wins will be ranked higher because they have a higher win percentage which makes sense.
- Each team is ranked based off of the number of times they have beaten another team. This is the first step to taking a teams schedule into account by ensuring that a team isn't ranked high simply because they played the worst team multiple times.
- In order to fully take into account a teams schedule, the ranking of a teams opponent is also taken into account. This ensures that a win against a good team has a more positive effect on their ranking than a win against a bad team.

B. Version 2

Version A of Keener's method still has room for improvement. The main way it could be improved is by taking point differentials into account. To do this, the entries of the preference matrix P will be changed. Instead of awarding one point per win, each team will be awarded the following:

$$p_{ij} = \frac{P_{ij} + 1}{T_{ij} + 2}$$

Where P_{ij} is the total points scored by team i across all their games against team j, and T_{ij} is the total number of points scored by both teams across all of the games between team i and j. The reason that 1 is added to the numerator and 2 is added to the denominator is because a close game ending in a score of 3-0 shows that the teams involved are very similar in rank. Without the addition of 1 in the numerator and 2 in the denominator, the winning team would receive a p_{ij} value of 1 which is the highest possible value, while the losing team would receive the same ranking as a team who was shutout 50-0. This will only be applied to teams who played each other. If team i never plays team j, $p_{ij} = 0$ so that a teams rank isn't altered by teams that they did not play.

To see how this result compares to the previous method, we must obtain the scores of each of the games. Suppose the games had the same outcomes, but now we know the score of each game, all of which is shown below.

Teams Playing	Winner	Score
Team 1 vs Team 2	Team 1	35-10
Team 1 vs Team 3	Team 3	21-17
Team 2 vs Team 3	Team 3	31-13
Team 1 vs Team 2	Team 1	38-14
Team 1 vs Team 3	Team 1	24-21
Team 2 vs Team 3	Team 2	17-14

The A matrix would be as follows:

$$A = \begin{bmatrix} 0 & 0.187 & 0.124 \\ 0.063 & 0 & 0.101 \\ 0.126 & 0.149 & 0 \end{bmatrix}$$

When plugged into MATLAB, the ranking vector is found to be:

$$\vec{r} = \begin{bmatrix} 0.657 \\ 0.433 \\ 0.617 \end{bmatrix}$$

When compared to the previous method, it is clear that the first method put more emphasis on wins while this method puts more emphasis on points. We can see this because the ranking of team 1 fell while the ranking of teams 2 and 3 rose because of the mostly close point differentials that occurred. It is unclear which method is more accurate because such a small data set was used. However, it is clear that the method developed by Keener is a beautiful application of the Perron-Frobenius theorem. There is also much left to be desired in terms of a method which balances wins and point differentials.

IV. Colley's Approach

As stated earilier, Colley's approach involves a Colley matrix C, ranking vector $\vec{r}$, and a vector $\vec{b}$. Colley approached the ranking of sports teams with win percentage in mind, so it makes sense that all of the elements in the equation $C\vec{r} = \vec{b}$ are filled using stats such as a teams wins, losses, total games played, ranking of the opponent, and number of games with a given opponent. Assuming a teams ranking is directly proportional to their win percentage, then the ranking of team i, r_i is defined as follows:

$$r_i = \frac{w_i}{t_i}$$

Where w_i is the amount of wins that team i has, and t_i is the total number of games that team i has played. Colley moved forward by recognizing the flaws of this ranking. For example, after the first game of the season, each team will have either won or lost (assuming there are no draws). This means that the teams who won will receive a ranking of 1 while the teams who lost will receive a ranking of 0. Colley reasoned that this wasn't a very good reflection, as the winning teams are not infinitely times better than the losing teams. He adjusted his method in a way similar to what was done to the preference matrix in Keener's method, by adding a one to the numerator of the ranking, and adding two to the denominator.

$$r_i = \frac{w_i + 1}{t_i + 2}$$

This subtle change improves the accuracy of the method, especially in the beginning of the season. Next, Colley wanted his ranking to take team i's schedule into account. To go about this, w_i was rewritten in the following way:

$$w_i = \frac{w_i - l_i}{2} + \frac{w_i + l_i}{2}$$

Where l_i is the number of losses that team i has. This can be simplified again to obtain:

$$w_i = \frac{w_i - l_i}{2} + \frac{t_i}{2}$$

From here, Colley notices that the second term, $\frac{t_i}{2}$ was equivalent to t_i times the average rank of all the teams ($\frac{1}{2}$). To truly factor in a teams schedule, he replaced this $\frac{1}{2}$ with the ranking of team i's opponent, r_j, which gave the following result:

$$w_i = \frac{w_i - l_i}{2} + \sum_{j=1}^{t_i} r_j$$

We can now use this new result for w_i and substitute it in to solve for the new r_i.

$$r_i = \frac{(\frac{w_i - l_i}{2} + \sum_{j=1}^{t_i} r_j) + 1}{t_i + 2}$$

To ensure that the system $C\vec{r} = \vec{b}$ always has a unique solution, c_{ij} and b_i will be defined accordingly:

$$c_{ij} = \begin{cases} t_i + 2 & i = j \\ -n_{ij} & i \neq j \end{cases} \qquad b_i = \frac{w_i - l_i}{2} + 1$$

Where n_{ij} is the total number games between team i and team j. To further understand this method, it will be applied to the same example data set that Keener's method was applied to. The outcomes of the games are again shown below:

Teams Playing	Winner
Team 1 vs Team 2	Team 1
Team 1 vs Team 3	Team 3
Team 2 vs Team 3	Team 3
Team 1 vs Team 2	Team 1
Team 1 vs Team 3	Team 1
Team 2 vs Team 3	Team 2

Using this data, Colley's method produces the following C matrix along with the following vector $\vec{b}$:

$$C = \begin{bmatrix} 6 & -2 & -2 \\ -2 & 6 & -2 \\ -2 & -2 & 6 \end{bmatrix} \qquad \vec{b} = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}$$

This system was plugged into MATLAB and it was found that:

$$\vec{r} = \begin{bmatrix} 0.625 \\ 0.375 \\ 0.5 \end{bmatrix}$$

We can see that team 1 will receive the highest ranking followed by team 3, and team 2 will receive the last ranking. This is the same end result that both versions of Keener's method reached, but the ranking vectors are different.

$$\vec{r_{KA}} = \begin{bmatrix} 0.724 \\ 0.340 \\ 0.601 \end{bmatrix} \qquad \vec{r_{KB}} = \begin{bmatrix} 0.657 \\ 0.433 \\ 0.617 \end{bmatrix} \qquad \vec{r_{C}} = \begin{bmatrix} 0.625 \\ 0.375 \\ 0.5 \end{bmatrix}$$

With this small of a data set, it is unclear which method is the most accurate representation of a teams ranking. What can be observed is that version A of Keener's method is the most rewarding to teams with a lot of wins, version B of Keener's method is rewarding to teams with high point differentials when they win and low point differentials when they lose, and Colley's method is more centered around win percentage, with the average ranking using Colley's method always being 0.5. Using either of Keener's methods on the example data set resulted in an average rank higher than 0.5. To get more of a full picture, a much larger, real data set will be analyze using all three of the described methods.

V. Numerical Results and Discussion

To test the accuracy of the three methods discussed, data from the 2020 NFL regular season will be analyzed. This data set was chosen because no other data was available in a format that could be easily converted into a file readable by MATLAB. Comparing data from NCAA basketball or football would yield a better comparison since they both have ranking committees in which methods similar to these, but far more complicated, are applied along with the committee's input to rank the teams. In fact, bcs rankings are the offical rankings of college football teams which involves the use of a Colley matrix which was defined in this report as C. Using data from NCAA football would've provided interesting comparison, however, there would have been far to much data to analyze given the size of this report.

In the NFL, there is no ranking committee, and teams are ranked off there records alone. Fourteen teams make the playoffs every year, and these teams are determined based on the top teams in each division. There are thirty-two teams in total which are split into two main divisions, the NFC and the AFC. From each division, seven teams make the playoffs, with the highest seed securing a first round bye.

To analyze which of the three methods discussed in this reoprt is most accurate, the scores and outcomes of the 2020 NFL regular season were analyzed. These rankings are shown below and include the top 21 teams so that all the teams who actually made the playoffs are shown. The ranking obtained using version A of Keener's method is denoted by r_{KA_i}. Similarly, r_{KB_i} represents the ranking determined by version B of Keener's method, and r_{C_i} is the ranking determined using Colley's method.

Rank	Team	r_{KA_i}
1	Bills	0.339
2	Chiefs	0.330
3	Seahawks	0.261
4	Rams	0.243
5	Steelers	0.235
6	Saints	0.223
7	Packers	0.219
8	Dolphins	0.205
9	Buccaneers	0.201
10	Browns	0.198
11	Titans	0.196
12	Cardinals	0.196
13	Raiders	0.193
14	Ravens	0.193
15	Colts	0.182
16	Patriots	0.163
17	Forty-Niners	0.159
18	Chargers	0.135
19	Redskins	0.131
20	Giants	0.122
21	Bears	0.117

Rank	Team	r_{KB_i}
1	Buccaneers	0.211
2	Saints	0.210
3	Chiefs	0.208
4	Rams	0.205
5	Packers	0.202
6	Ravens	0.200
7	Bills	0.200
8	Seahawks	0.194
9	Dolphins	0.193
10	Cardinals	0.190
11	Steelers	0.189
12	Colts	0.184
13	Panthers	0.183
14	Falcons	0.180
15	Titans	0.180
16	Forty-Niners	0.179
17	Bears	0.179
18	Raiders	0.171
19	Redskins	0.170
20	Browns	0.168
21	Vikings	0.166

Rank	Team	r_{KC_i}
1	Chiefs	0.394
2	Bills	0.349
3	Packers	0.284
4	Steelers	0.269
5	Seahawks	0.252
6	Saints	0.243
7	Browns	0.195
8	Ravens	0.191
9	Titans	0.186
10	Buccaneers	0.185
11	Rams	0.180
12	Colts	0.179
13	Dolphins	0.124
14	Raiders	0.0644
15	Cardinals	0.055
16	Bears	0.006
17	Patriots	-0.006
18	Redskins	-0.021
19	Chargers	-0.032
20	Vikings	-0.036
21	Forty-Niners	-0.051

It is reasonable to assume that a teams ranking is directly correlated to how good that team is. This means that if two teams face each other, the team with the higher ranking has a higher probability of winning. Of course the team with the higher probability of winning doesn't always win, but this concept is still mathematically sound. To compare the accuracy of the three methods, each method will be used to predict the outcomes of all the playoff games, in which the team with the higher ranking is predicted to win. When this is done, version A of Keener's method correctly predicted six of the thirteen games, version B of Keener's method correctly predicted ten of the games, and Colley's method predicted eight of the games. Out of the four teams who made it to the AFC and NFC championship games (the semifinals), version A of Keener's method only had one of those teams in its top four, version B had two of those teams in its top four while also predicting the teams in the Super Bowl and the winner, and Colley's method also predicted two of the last four teams standing. From this it is clear that version B of Keener's method was the most accurate method when applied to this specific data set, with Colley's method as a close second.

What can be learned from this is that point differentials, along with win percentage, are a very good indicator of how good a team actually is. As stated earlier, when college teams are ranked, these things are taken into consideration, but NFL teams are ranked solely based off of their records. The NFL is traditional and likely won't change how teams are ranked, but the methods covered in this report could still be useful in predicting the outcomes of games.

VI. Conclusion

This report clearly shows the power of linear algebra and its applications. The methods covered in this report only scratch the surface of what is possible when it comes to ranking sports teams using mathematics. The results obtained when analyzing the 2020 NFL season show strong evidence to support the accuracy of version B of Keener's method specifically. This method not only correctly predicted ten of the thirteen playoff games correctly, but also which teams would be in the Super Bowl and which team would win. The Tampa Bay Buccaneers won the Super Bowl and based off their record, they were the fifth seed out of seven on their side of the bracket. This shows how a teams record isn't always a good representation of how good they are. When version B of Keener's method was applied, the Buccaneers were ranked number one overall which shows that they had a difficult schedule along with point differentials that reflected a good team.

Clearly Colley's method also shows potential since it is the base on which the bcs rankings (the rankings used in NCAA football) were founded on. To create an even more accurate method for ranking sports teams, Colley's method and version B of Keener's method could be combined. This would ensure that win percentage and point differentials would both be accounted for. This combination could result in a more complex system such as a system without a solution in which the least squares solution could be found using the pseudo inverse of the matrix. Other concepts from Matrix Methods could also be applied in the realm of ranking sports teams.

VII. References

- *The Perron-Frobenius theorem and the ranking of football teams*, by JP Keener, SIAM Review (1993).
- *The Numerical ranking of sports teams*, by Mar McClure, https://www.marksmath.org/visualization/eigenbrackets/
- *Colley's Method*, by Wes Colley, https://www3.nd.edu/ apilking/Math10170/Information/Lectures
- *NFL-Regular Season Complete* http://www.playoffstatus.com/nfl/nflschedule.html (2020)
- *2020-21 NFL playoffs*, https://en.wikipedia.org/wiki/2020
- This link can be used to download the NFL data used in this project: https://sites.google.com/view/matmethprojdata/data

VIII. Appendix

```
%Nick Larson

clear;
clc;

%Least squares example in Background
A0=[1 -1 0;0 1 -1;-1 0 1];
b0=[14;3;6];
r0=lsqr(A0,b0);

lsqr converged at iteration 1 to a solution with relative residual
 0.86.

%Code for example in Version 1 of Keener's Method
A1=[0 1/2 1/4;0 0 1/4;1/4 1/4 0];
[evec1,eval1]=eig(A1);

%Convert diagnal matrix eval1 to a vector with the eigenvalues
eval11=zeros(1,length(eval1));
for i=1:length(eval1)
   eval11(1,i)=eval1(i,i);
end

%Find the max eigenvalue and corresponding eigenvector (ranking
 vector)
lambda1=max(eval11);

for i=1:length(eval11)
   if eval11(1,i)==lambda1
       ind1=i;
   end
end

r1=evec1(:,ind1);

%Ensure r1 has all positive components instead of all negative
 components
if r1(1)<0
    r1=r1*-1;
end

%Code for example in Version 2 of Keener's Method
A2=[0 0.1869 0.1235;0.0631 0 0.1006;0.1264 0.1493 0];
[evec2,eval2]=eig(A2);

%Convert diagnal matrix eval2 to a vector with the eigenvalues
eval22=zeros(1,length(eval2));
for i=1:length(eval2)
   eval22(1,i)=eval2(i,i);
end

%Find the max eigenvalue and corresponding eigenvector (ranking
 vector)
```

```
lambda2=max(eval22);

for i=1:length(eval22)
   if eval22(1,i)==lambda2
       ind2=i;
   end
end

r2=evec2(:,ind2);

%Ensure r2 has all positive components instead of all negative
 components
if r2(1)<0
    r2=r2*-1;
end

%Example for Colley's method
C=[6 -2 -2;-2 6 -2;-2 -2 6];
b=[2;0;1];
r3=linsolve(C,b);

%Load Data for 2020 NFL season to do a comparison with all three
 methods
Table=readtable('MatMethodsProjData.csv');
Table=Table(:,2:5);
Data=table2cell(Table);

%Create a mat with just the scores
ScoresMat=zeros(length(Data),2);
for i=1:length(Data)
   ScoresMat(i,1)=Data{i,2};
   ScoresMat(i,2)=Data{i,3};
end

%Ranking of NFL teams using Keener version 1

%Assign teams a number 1-32
TeamsVec=cell(32,1);
for i=1:16
   TeamsVec(i)=Data(i,1);
   TeamsVec(i+16)=Data(i,4);
end

%Replace names of team with associated number
DataTeams=zeros(256,2);
for j=1:2
    for i=1:256
        for k=1:32
            if j==1
                if length(Data{i,1})==length(TeamsVec{k})
                    if Data{i,1}==TeamsVec{k}
                        DataTeams(i,1)=k;
                    end
                end
```

```
            else
                if length(Data{i,4})==length(TeamsVec{k})
                    if Data{i,4}==TeamsVec{k}
                        DataTeams(i,2)=k;
                    end
                end
            end
        end
    end
end

%Combine DataTeams and ScoresMat
DataMat=[DataTeams(:,1) ScoresMat(:,1) ScoresMat(:,2) DataTeams(:,2)];

%Create A Matrix
AK1=zeros(32);
for i=1:256
    if DataMat(i,2)>DataMat(i,3)

 AK1(DataMat(i,1),DataMat(i,4))=AK1(DataMat(i,1),DataMat(i,4))+1;
    end
    if DataMat(i,3)>DataMat(i,2)

 AK1(DataMat(i,4),DataMat(i,1))=AK1(DataMat(i,4),DataMat(i,1))+1;
    end
end
AK1=AK1/16;

%Find evals and evecs
[evecK1,evalK1]=eig(AK1);

%Convert evalK1 to a vector
evalK11=zeros(1,length(evalK1));
for i=1:length(evalK1)
    evalK11(i)=evalK1(i,i);
end

%Find max eigenvalue and corresponding eigenvector
lambdaK1=max(evalK11);

for i=1:length(evalK11)
   if evalK11(i)==lambdaK1
       indK1=i;
   end
end

rK1=evecK1(:,indK1);

%Ensure ranking vector is positive
if rK1(1)<0
    rK1=rK1*-1;
end

%Sort teams from highest to lowest ranking
```

```
sortedrK1=sort(rK1,'descend');
rK1Matsorted=zeros(32,2);
rK1Matsorted(:,2)=sortedrK1;

for i=1:32
    for j=1:32
        if sortedrK1(i)==rK1(j)
            rK1Matsorted(i,1)=j;
        end
    end
end

FinalrK1=cell(32,3);

for i=1:32
    FinalrK1{i,1}=i;
    FinalrK1{i,3}=rK1Matsorted(i,2);
    FinalrK1{i,2}=TeamsVec(rK1Matsorted(i,1));
end

FinalrK1=cell2table(FinalrK1);
FinalrK1.Properties.VariableNames={'Rank','Team','r_ij'};

%Ranking NFL teams using Keener version 2

%Create A matrix
numAK2=zeros(32);
for i=1:256
    for j=1:32
       if DataMat(i,1)==j

 numAK2(DataMat(i,1),DataMat(i,4))=numAK2(DataMat(i,1),DataMat(i,4))+DataMat(i,2);
       end
       if DataMat(i,4)==j

 numAK2(DataMat(i,4),DataMat(i,1))=numAK2(DataMat(i,4),DataMat(i,1))+DataMat(i,3);
       end
    end
end

denAK2=zeros(32);
for i=1:256
    for j=1:32
        if DataMat(i,1)==j

 denAK2(DataMat(i,1),DataMat(i,4))=denAK2(DataMat(i,1),DataMat(i,4))+DataMat(i,3)+
        end
        if DataMat(i,4)==j

 denAK2(DataMat(i,4),DataMat(i,1))=denAK2(DataMat(i,4),DataMat(i,1))+DataMat(i,3)+
        end
    end
end
```

```
%Add one to numerator and two to denominator
for i=1:32
    for j=1:32
        if numAK2(i,j)>0
           numAK2(i,j)=numAK2(i,j)+1;
        end
           denAK2(i,j)=denAK2(i,j)+2;
    end
end

%Form AK2 mat
AK2=zeros(32);
for i=1:32
    for j=1:32
        AK2(i,j)=numAK2(i,j)/denAK2(i,j);
    end
end
AK2=AK2/16;

%Find evals and evecs
[evecK2,evalK2]=eig(AK2);

%Convert evalK1 to a vector
evalK22=zeros(1,length(evalK2));
for i=1:length(evalK2)
    evalK22(i)=evalK2(i,i);
end

%Find max eigenvalue and corresponding eigenvector
lambdaK2=max(evalK22);

for i=1:length(evalK22)
   if evalK22(i)==lambdaK2
       indK2=i;
   end
end

rK2=evecK2(:,indK2);

%Ensure ranking vector is positive
if rK2(1)<0
    rK2=rK2*-1;
end

%Sort teams from highest to lowest ranking
sortedrK2=sort(rK2,'descend');
rK2Matsorted=zeros(32,2);
rK2Matsorted(:,2)=sortedrK2;

for i=1:32
    for j=1:32
        if sortedrK2(i)==rK2(j)
            rK2Matsorted(i,1)=j;
        end
```

```
    end
end

FinalrK2=cell(32,3);

for i=1:32
    FinalrK2{i,1}=i;
    FinalrK2{i,3}=rK2Matsorted(i,2);
    FinalrK2{i,2}=TeamsVec(rK2Matsorted(i,1));
end

FinalrK2=cell2table(FinalrK2);
FinalrK2.Properties.VariableNames={'Rank','Team','r_ij'};

%Ranking of NFL teams using Colley's method

%Create C matrix
C=zeros(32);

AK1ref=AK1*16;

for i=1:32
    for j=1:32
        if i==j
            C(i,j)=18;
        else
            C(i,j)=-AK1ref(i,j);
        end
    end
end

%Create b1 vec
bC=zeros(32,1);
wC=zeros(32,1);
lC=zeros(32,1);

for i=1:256
        if DataMat(i,2)>DataMat(i,3)
            wC(DataMat(i,1))=wC(DataMat(i,1))+1;
            lC(DataMat(i,4))=lC(DataMat(i,4))+1;
        end
        if DataMat(i,3)>DataMat(i,2)
            wC(DataMat(i,4))=wC(DataMat(i,4))+1;
            lC(DataMat(i,1))=lC(DataMat(i,1))+1;
        end
end

for i=1:32
    bC(i)=((wC(i)-lC(i))/2)+1;
end

%Solve for ranking vector
rC=linsolve(C,bC);
```

```
%Sort teams from highest to lowest ranking
sortedrC=sort(rC,'descend');
rCMatsorted=zeros(32,2);
rCMatsorted(:,2)=sortedrC;

for i=1:32
    for j=1:32
        if sortedrC(i)==rC(j)
            rCMatsorted(i,1)=j;
        end
    end
end

FinalrC=cell(32,3);

for i=1:32
    FinalrC{i,1}=i;
    FinalrC{i,3}=rCMatsorted(i,2);
    FinalrC{i,2}=TeamsVec(rCMatsorted(i,1));
end

FinalrC=cell2table(FinalrC);
FinalrC.Properties.VariableNames={'Rank','Team','r_ij'};
```

Published with MATLAB® R2021a