Strength Of Victory Based Ranking System

Joshua L. Padgett
Math 164: Scientific Computing

May 3, 2004

Abstract

Current sports ranking systems are mostly based on wins, losses and strength of schedule. In this paper I devise a formula to rank a group of sports teams based entirely on Margin of Victory. I introduce Strength of Victory and Expected Strength of Victory and derive a formula for ranking a group of teams based on those criteria. I argue that this formula is comparable to other methods which are based on different information. In addition, I argue that it ranks the teams fairly.

1 Introduction

The interest of this project is in ranking a group of competitors. In general these competitors do not all compete against one another, i.e. it is not a round-robin competition. Only in special cases of this problem would it be round-robin. Thus, there is an information problem, where we are ranking teams against each other when they haven't played each other, and may not even share a single opponent. I am interested in the general case of this problem, where a single model can accurately rank any group, regardless of the type of competition.

Interest in this subject is varied. Recently, there was much interest in the the Bowl Championship Series ranking system stemming from the College Football season and its difficulties in picking the two teams to play for the 2003 National Championship. Controversy over the BCS system was rampant, as the human polls voted the top 3 teams differently than the BCS

computed them [2]. Other interest stems from determining at-large bids into tournaments, where some competitors are automatically entered by winning a previous competition but the field of teams must be completed from the rest of the possible teams. An example of this is the NCAA basketball tournament, and Division III playoffs.

Thus, given a set of teams, with a season played among those teams, can we rank the teams from best to worst? If so does this method compare favorably to other methods?

2 Common Terms

For the purpose of this project we will understand each *team* to be a separate entity, a *game* to mean a competition between two teams, for which a score is recorded, and a *season* to be a collection of these games. There need not be a game played between each team (*round-robin*), in fact, we are more interested in the case when there are many teams who do not play each other.

3 Factors in Determining Rank

The following is a list of factors that contribute to a ranking system, it may not be exhaustive, but the most important factors are certainly in this list:

- Win/Loss Record
- Margin of Victory (MOV)
- Strength of Schedule (SOS)
- Location of Contest
- Human Factors
  - Prestige
  - Perception
  - Bias

Win/Loss Record is self-explanatory. Margin of Victory is the amount by which a team beats its opponent. The strength of a team's schedule is a measure of the competition it has faced during the season. There is no set standard for computing the strength of schedule factor. Some ranking systems take into account the location of a game, assuming the home team has an advantage over the visiting team. Obviously in any human poll type ranking system the human factors weigh heavily in the rankings. How else would pre-season polls be created? Even in some mathematical ranking systems the human factors are inherent through *pre-rankings*. These systems require the teams to be pre-ranked, and thus the human factors weigh in.

Generally Win/Loss Record is considered the most important factor. There are ranking systems that rely solely on Win/Loss Record and SOS, for example the Colley Ranking System [1]. To gain some perspective, the BCS has determined that margin of victory should not be a factor in their ranking system, and uses a combination of SOS, Win/Loss records, and human polls. I don't believe this to be a fair assessment. The argument for not using Margin of Victory is that it rewards teams that beat inferior opponents handily. In a purely mathematical ranking system, this should be an addressable issue, and thus, my system uses MOV.

4 Introduction of Components

I will introduce several terms now, which will be descriptions of portions of the formula used for the rankings. The *Strength of Victory (SOV)* is an indication of how well a team performed in a given game. Ideally in any given game we would be able to signify what the outcome of that game means. By this I mean it is easy to imagine high-scoring games having large margin's of victory, and low-scoring games having small margin's of victory. It follows then, that the margin of victory in a low-scoring game is more important than the margin of victory in a high-scoring game. We want any formula for Strength of Victory to encompass this fact. In addition, the formula should be symmetric, since the previous discussion applies to both the winning and losing teams.

We are also interested in the fact that prior to a contest the two teams are

not equal competitors. To account for this fact we introduce the *Expected Strength of Victory (ESOV)*. This formula will be an indication of what the outcome of a game was supposed to be. The ESOV is based on the difference between the rankings of the two teams competing.[4]

5 The Formula

To summarize the attributes we wish the SOV component to have:

- Decreasing value for increasing total score
- Increasing value for increasing margin
- Decreasing increase in value for increasing margin
- Symmetric for winning and losing team.

Assume S_i, S_j are the score for teams i and j, respectively, in a game between the two. The following formula lends itself very nicely to the previous desired attributes:

$$SOV = (S_i - S_j)/(S_i + S_j)$$

As one can see the formula is symmetric. One could verify the other attributes by graphing the formula in 3 space, or taking the derivative of the formula.

The other component, then, is the ESOV. Lets assume we have ranking's for each of the teams R_i for each team i. Then in a competition between two teams, we expect the margin of victory to be a factor of the average margin of victory, depending on the difference in quality between the two teams. The criterion I use is the standard deviation of the difference in rankings (StdDev_R). Thus, we define the Expected Margin of Victory for team i against team j to be

$$ESOV_{ij} = AvSOV * (R_i - R_j)/\text{stdDev}_R$$

To compute the change in ranking of team i in playing team j we compute the difference between the SOV and ESOV

$$dR_i = SOV_{ij} - ESOV_{ij}.$$

To find the change in ranking over the whole season, we would then iterate over the whole season for team i playing g games

$$dR_{ig} = \sum_{j=1}^{g} SOV_{ij} - ESOV_{ij}$$

We can compute the change in each team's rankings in the same way.

6 Results

I decided to test my ranking system on the 2003 NCAA Division I College football season, up to the week of December 6, 2004. The results can be found in Table 1. I chose this data for two reasons: it was available, and there was much controversy over who the top two teams should be. The human polls ranked Southern California (USC) first, Louisiana State (LSU) second and Oklahoma third, but the BCS ranked Oklahoma first, LSU second and USC third.

7 Understanding the Rankings

Some conclusions can definitely be drawn from Table **??**. Since the ranking system is based solely on Margin of Victory, I looked at the scores over the season of the teams that seemed out of place, i.e. didn't correlate to the other ranking systems. I also looked at how other ranking systems ranked these teams. I will use Kansas State and California as examples from the Division I rankings.

Both Kansas State and California ranked much higher using my formula than in the other ranking systems. I think there are two reasons for this: they both beat very highly ranked teams, and their margins were small in losses and large in wins. Kansas State beat Oklahoma (handily), the number one ranked team, and California beat Southern California. I believe this ranking system rewards teams that win against much higher ranked opponents because of the standard deviation portion of the formula. Virginia is another example of this trend, as they beat a highly ranked Florida State team.

Table 1: Comparative Rankings for NCAA Div I College Football
Top 25 Ranked Teams by SOV formula, for games up to December 6,2003

		Massey	BCS	Colley	Coaches	Assoc. Press	Consensus
1	Oklahoma	2	1	1	3	3	1
2	LouisianaState	1	2	2	2	2	2
3	KansasState	8	10	12	10	8	8
4	Michigan	4	4	6	4	5	4
5	SouthernCalifornia	3	3	3	1	1	3
6	Texas	6	6	8	5	4	5
7	Georgia	5	12	11	11	11	11
8	FloridaState	9	7	5	8	9	6
9	Miami(Ohio)	7	11	9	15	14	7
10	BoiseState	13	17	16	16	18	14
11	Miami(Florida)	11	9	7	9	10	10
12	OhioState	12	5	4	6	7	9
13	Tennessee	10	8	10	7	6	12
14	Nebraska	17	20	18	21	22	17
15	Minnesota	24	25	28	20	24	23
16	Utah	23	22	14	25	25	21
17	Iowa	14	13	15	12	13	13
18	Mississippi	18	19	24	18	21	20
19	WashingtonState	19	16	19	14	15	19
20	NewMexico	37	45	39	-	-	41
21	Florida	15	15	23	17	17	16
22	California	9	43	44	-	-	39
23	Virginia	34	44	42	-	-	40
24	Purdue	15	15	17	13	12	15
25	Arkansas	24	38	26	-	-	24

8 Conclusions

It is clear this ranking system has some bias towards certain types of victories. This can be seen by looking at the resulting rankings, and also by inspecting the formula itself. Some of the conclusions that can be drawn from this investigation include:

- Large Margins of Victory mean much more in low-scoring games than high-scoring games

- Lower-ranked teams gain substantially by beating highly ranked teams
- Losing games by small amounts against good teams reduces the penalty for losing
- Stiffer overall competition leads to a higher ranking

Overall, I was pleased with the correlation between my rankings and the other rankings. Considering the fact that this system is based solely on Margin of Victory, it is interesting that Strength of Schedule is inherently taken into account. Although the formula is relatively simple I think qualitatively it does rather well.

Quantitatively, I must confess the system has some flaws. The best victory in this system is holding a team scoreless. No matter what the winning team's score is, holding an opponent scoreless results in an SOV of 1. This allows teams to gain ranking points in an otherwise uncompetitive game.

Using standard deviation as a measure of comparison enhances outlier results. Since a majority of the difference in rankings are going to be within one standard deviation, and almost all are within two standard deviations, the few outlier games beyond those bounds can have magnified consequences. This can be directly seen in the high rankings of California and Virginia.

In conclusion, I think a ranking system based on Margin of Victory is certainly feasible, and can answer the question of ranking a season of sports competition. I believe this study showed this to be true. I also believe this system would work well beyond college football. Large amounts of data are not available for many other sports, though, so this claim can not be verified.

9 Future Modifications

The Strength of Victory formula is good, but not perfect. A variation using different exponent powers for the numerator and denominator would be a good place to start that investigation. The numerator should be raised to an odd power, so as to keep the results symmetric.

I believe using the standard deviation of the difference of the rankings is a very strong indicator of the expectations of a given game. There are other statistical methods for dealing with these types of issues which could be investigated.

Another area of interest would be to investigate time dependent margin of victory formulas. In fact, I looked into this idea, but it favored the end of the season too heavily, in my opinion. I have included the code for the time dependent algorithm, for interested parties.

10 Acknowledgements

I made extensive use of the college football scores on James Howell's web page [6]. Also, Kenneth Massey's page was invaluable for comparing my ranking system [5].

11 Disclaimer

This ranking system is entirely original as far as the author knows. Any resemblance to other ranking systems is purely coincidental. All the terms introduced were devised by the author, as were the formulas, unless otherwise noted. Since the formulas are not extremely complicated, it is not hard to imagine them being implemented previously in some other ranking system. It is impossible to look at every ranking system ever devised to ensure that this is an original ranking system, and it is not feasible under the time constraints of this project to have done so.

References

[1] Wesley N. Colley. *Colley's Bias Free College Football Ranking Method: The Colley Matrix Explained*. Princeton University, 2002.

[2] *BCS Explained*. `http://www.collegefootballpoll.com/bcs_explained.html`.

[3] Kenneth Massey. *M*assey Rating Description. 1999.

[4] Martin Avila. *B*rief Description of the EMV Rating System.

[5] *College Football Ratings*. `http://www.masseyratings.com/cf/`.

[6] *College Football Scores*. `http://www.cae.wisc.edu/~dwilson/rsfc/history/howell/`.

Appendix

```
%function createMatrix
% createMatrix takes a list of scores, one game per line
% and creates a matrix of the scores.
% Each team has a row and column.
% A team's own scores go in their column.
% A team's opponents' scores go in the opponents column of the team's row.
% Ex. If team i vs team j, team i's score goes in cell[j,i] (row j, column i)
%                          team j's score goes in cell[i,j] (row i, column j).
% A non-game is scored 1-1.
%
% file is the name of the file to take input from.
% Each line of input must look exactly like this
%
%    TeamA # TeamB #
% One game per line.
% The team names can have no spaces.
%

% Scores is a cell array containing each round of scores
% GameIndex tells which games have been played, and also tells
%  who won each game
% games is the number of games
% teamnum is the number of teams
% Teams is a struct with the team names and assigned team numbers

function [Scores,GamesIndex,gamesR,teamnum,Teams] = createMatrixTI(file)

% t1, list of teams
% s1, list of scores for those teams
% t2, list of opponents
% s2, list of scores for those opponents

teamnum = 1;


[t1,s1,t2,s2] = textread(file,'%s %n %s %n');
```

```
games = length(t1);
gamesR(1) = 0;

Teams = struct(t1{1},teamnum);

for i=1:length(t1)

    if(~isfield(Teams,t1{i}))
        teamnum = teamnum+1;
        Teams = setfield(Teams,t1{i},teamnum);
    end
    if(~isfield(Teams,t2{i}))
        teamnum = teamnum+1;
        Teams = setfield(Teams,t2{i},teamnum);

    end
end

Scores = cell(1);
Scores{1} = ones(teamnum);
GamesIndex = cell(1);
GamesIndex{1} = zeros(teamnum);

for i=1:games
    round = 1;
    a = getfield(Teams,t1{i});
    b = getfield(Teams,t2{i});

    notdone = 1;
    while(notdone)
        if(round > length(Scores))
            Scores{round} = ones(teamnum);
            GamesIndex{round} = zeros(teamnum);
            gamesR(round) = 0;
            notdone = 0;
        end
        if(GamesIndex{round}(a,b) == 0)
```

```
            Scores{round}(b,a) = s1(i);
            Scores{round}(a,b) = s2(i);
            GamesIndex{round}(b,a) = 1 -2*(s1(i)<s2(i));
            GamesIndex{round}(a,b) = 1 -2*(s2(i)<s1(i));
            gamesR(round) = gamesR(round)+1;
            notdone = 0;
        else
            round = round+1;
        end
    end
end


function rankSeason(its,fromfile,tofile)

[S,GI,games,teams,Teams] = creatematrixTI(fromfile);
% S stands for scores
%S has a row and column for every team.
%A teams own scores go in their column.
%A teams opponents scores go in their row.
%Ex. If team i vs team j, team i's score goes in cell[j,i]
%                         team j's score goes in cell[i,j]
% A non-game is scored 1-1.

TotMgn = 0;
TotScores = 0;

Ranks = zeros(1,teams); % Vector of team's ranks
Games = zeros(1,teams); % Vector of team's games played
Diffs = zeros(teams);   % A matrix containing the difference in ranks

rounds = length(S);

SOV = cell(1,rounds);
GamesIndices = cell(1,rounds);
```

```
for r=1:rounds
    TotScores = TotScores + sum(sum(S{r})) - teams - 2*games(r);
    TotMgn = TotMgn + sum(sum(abs(S{r}-S{r}.')))/2;
    AvStrength = TotMgn/TotScores;

    %OwnScore - OppScore/TotalScoreOfGame
    SOV{r} = ((S{r}-S{r}.')./(S{r}+S{r}.'));
    GamesIndices{r} = abs(GI{r});

end

% GIs is a matrix denoting any game thats been played.
GIs = GamesIndices{1};
for r=2:rounds
    GIs = or(GIs,GamesIndices{r});
end

for i=1:its

        for j=1:teams


            Diffs(:,j) = Ranks(j)*GIs(:,j)...
                    - (Ranks.*GIs(j,:)).';
        end
        stddev = std_dev(Diffs)

        if(stddev ~= 0)
            for r=1:rounds
                Ranks = Ranks + sum(SOV{r}...
                    - (AvStrength/stddev)*(Diffs.*GamesIndices{r}));
            end
        else
            for r=1:rounds
                Ranks = Ranks + sum(SOV{r});

            end
        end
```

```
    end
end

[Y,Order] = sort(Ranks);
ts = fieldnames(Teams);

fid = fopen(tofile,'w');

for i=teams:-1:1
    fprintf(fid,'%i & %s &\\\\ \n',teams-i+1,ts{Order(i)});
end

fclose(fid);

%function createMatrix
% createMatrix takes a list of scores, one game per line
% and creates a matrix of the scores.
% Each team has a row and column.
% A team's own scores go in their column.
% A team's opponents' scores go in the opponents column of the team's row.
% Ex. If team i vs team j, team i's score goes in cell[j,i] (row j, column i)
%                          team j's score goes in cell[i,j] (row i, column j).
% A non-game is scored 1-1.
%
% file is the name of the file to take input from.
% Each line of input must look exactly like this
%
%    TeamA # TeamB #
% One game per line.
% The team names can have no spaces.
%

% Scores is a cell array containing each round of scores
% GameIndex tells which games have been played, and also
% tells who won each game
% games is the number of games
% teamnum is the number of teams
% Teams is a struct with the team names and assigned team numbers
```

```
function [Scores,GamesIndex,gamesR,teamnum,Teams] = createMatrixTD(file)

% t1, list of teams
% s1, list of scores for those teams
% t2, list of opponents
% s2, list of scores for those opponents

teamnum = 1;



GC = cell(4,length(file));
for k=1:length(file)

    [GC{1,k},GC{2,k},GC{3,k},GC{4,k}] =...
        textread(file{k},'%s %n %s %n');

    gamesR(k) = length(GC{1,k});

    if(k==1)
        Teams = struct(GC{1,k}{1},teamnum);
    end

    for i=1:length(GC{1,k})

        if(~isfield(Teams,GC{1,k}{i}))
            teamnum = teamnum+1;
            Teams = setfield(Teams,GC{1,k}{i},teamnum);
        end
        if(~isfield(Teams,GC{3,k}{i}))
            teamnum = teamnum+1;
            Teams = setfield(Teams,GC{3,k}{i},teamnum);
        end
    end
end

Scores = cell(1,k);
```

```
GamesIndex = cell(1,k);

for k=1:length(file)

    Scores{k} = ones(teamnum);
    GamesIndex{k} = zeros(teamnum);

    for i=1:gamesR(k)

        a = getfield(Teams,GC{1,k}{i});
        b = getfield(Teams,GC{3,k}{i});

        if(GamesIndex{k}(a,b) == 0)
            Scores{k}(b,a) = GC{2,k}(i);
            Scores{k}(a,b) = GC{4,k}(i);
            GamesIndex{k}(b,a) = 1 -2*(GC{2,k}(i)<GC{4,k}(i));
            GamesIndex{k}(a,b) = 1 -2*(GC{4,k}(i)<GC{2,k}(i));
        else
            sprintf('%s %s %s',...
        'Error: Teams can not play each other twice in the same round',...
                GC{1,k}{i},GC{3,k}{i})

        end

    end
end

function RoundRanks = rankSeasonTimeDep(its,fromfile,tofile)

[S,GI,games,teams,Teams] = creatematrixTD(fromfile);
% S stands for scores
%S has a row and column for every team.
%A teams own scores go in their column.
%A teams opponents scores go in their row.
%Ex. If team i vs team j, team i's score goes in cell[j,i]
%                         team j's score goes in cell[i,j]
% A non-game is scored 1-1.
```

```
rounds = length(S);

TotMgn = 0;
TotScores = 0;
RoundRanks = cell(1,rounds);
Ranks = zeros(1,teams); % Vector of team's ranks
Games = zeros(1,teams); % Vector of team's games played
Diffs = cell(1,rounds);

SOV = cell(1,rounds);
GamesIndices = cell(1,rounds);

for r=1:rounds
    TotScores = TotScores + sum(sum(S{r})) - teams - 2*games(r);
    TotMgn = TotMgn + sum(sum(abs(S{r}-S{r}.')))/2;


    %OwnScore - OppScore/TotalScoreOfGame
    SOV{r} = ((S{r}-S{r}.')./(S{r}+S{r}.'));
    GamesIndices{r} = abs(GI{r});

end
AvStrength = TotMgn/TotScores;
% GIs is a matrix denoting any game thats been played.
%GIs = GamesIndices{1};
%for r=2:rounds
 %   GIs = or(GIs,GamesIndices{r});
 %end



for r=1:rounds

    Diffs{r} = zeros(teams);   % A matrix containing the difference in ranks
    for i=1:its
        for j=1:teams

            Diffs{r}(:,j) = Ranks(j)*GamesIndices{r}(:,j)...
```

```
                    - (Ranks.*GamesIndices{r}(j,:)).';
        end
        stddev = std_dev(Diffs{r})

        if(stddev ~= 0)
            Ranks = Ranks + sum(SOV{r}...
                    - (AvStrength/stddev)*(Diffs{r}.*GamesIndices{r}));

        else
            Ranks = Ranks + sum(SOV{r}) ;

        end
    end
    RoundRanks{r} = Ranks;
end

[Y,Order] = sort(Ranks);
ts = fieldnames(Teams);

fid = fopen(tofile,'w');

for i=teams:-1:1
    fprintf(fid,'%i & %s &\\\\ \n',teams-i+1,ts{Order(i)});
end

fclose(fid);
```