CS 229 Final Project Football Ranking System Lawrence Wisne

I. Introduction

In this paper I will describe the techniques that I used to create a simple but efficient ranking system for college football teams. To begin, I will go over the methods that are used in ranking systems that divulge their methodology. There are a multitude of erroneous assumptions made by other ranking programs that my system works to avoid. I will then go over the methods that I have employed to improve on previous ranking attempts. The overall goal of this project was to create a ranking system which can be used by the Bowl Championship System (BCS). Currently, the rankings used by the BCS to determine the participants in highly lucrative bowl games are determined partly by the result of six different computer rankings. Some of the rankings used make the kind of arbitrary assumptions that should, in my opinion, be avoided by a computer-based ranking system.

II. Overview of Current Systems

First and foremost, current systems that are employed use the "politically correct" assumption that the final score of a game doesn't matter and just the winner/loser of the game is relevant. In other words, the algorithms don't get to know whether the game is a 1-point victory or a 40-point blowout. I am very sure that this methodology is incorrect. Previous attempts at capturing the value of margin of victory have used a linear curve to view point differentials. Assuming a 21-point victory is 21 times better than a 1-point victory is just as wrong as not using the score at all, which is why most programs have not suffered much of a lack of performance by ignoring the margin of victory.

A second flaw of most computer ranking systems is the one that directly correlates to the content of this course. Each ranking system has its own variety of features that it uses to determine the rank of teams. In most cases, the main differentiating feature that separates programs' rankings is strength of schedule calculation; however, many systems use other features as well. The common bond between these features is that they are weighted arbitrarily. The author of the program has decided ahead of time what is important and what isn't important, and uses these assumptions to weight the features that determine a ranking. Another goal of mine when implementing my solution is to avoid arbitrary decisions when calculating a set of predictions and the rankings that go along with those predictions.

III. Inspiration Behind This Ranking Algorithm

The algorithm I decided to use in ranking teams originated from the idea presented in problem 2 of PS4. In that problem, conference attendees submit paper proposals and the papers are reviewed by a set of reviewers. Each reviewer gives a rating that estimates the quality of a paper. However, it is natural that each reviewer uses a different set of criteria when evaluating papers, and additionally the prevalence of a certain ranking from a certain reviewer changes for each reviewer. Some reviewers may

only give the highest score to the very best paper read, or may never give it at all. In contrast, other reviewers may give the highest score to any paper which he or she deems worthy of presentation at the conference. As such, one can say that the ratings for each reviewer are systematically biased around some mean value with a certain variance.

Upon inspection of the problems presented in the domain of college football scores and the goal of this project, I realized that this "paper/review" problem is analogous to the one I was trying to solve. For each game played by a football team, one could say that the outcome of the game is a "review" of both the offensive and defensive prowess of the team. In addition, these reviews are biased in the same way as those in the paper/review problem, as each team has a systematic bias in the results it yields.

A simple example found in real data is that USC tends to score a very high number of points in each game. If we treat the number of points team i scores against team j as a review of the defensive ability of team j, then USC's "reviews" of the defensive ability of each team on its schedule are very largely biased towards high numbers. Since this is the case, then any team which holds USC to a low number of points should receive a large amount of credit for this accomplishment. This is directly analogous to being the only paper rated highly by a reviewer that otherwise gave low-to-mediocre ratings.

It should be noted that this approach to solving this problem is a fairly large departure from the methods I had planned on using earlier in the quarter. My original plan was to use a methodology that is quite common and domain specific, where I set some number of features for each team and learn on those features. Where I had hoped to create better results by learning features efficiently, I felt that this idea would create results that are quite different from, and hopefully superior to, other systems I have researched while coming up with ideas for this project.

IV. The Ranking Algorithm and its Derivation

Taking the paper/review problem as a guide, I set out to create a system which utilized the EM class of algorithms in order to solve my football ranking problem. However, there were some tractibility problems encountered along the way which required a significant simplification of the ideas used to solve the paper/review problem. With this in mind, I will give the original ideas and explain what led to the design decision to simplify them.

If implemented as an EM Algorithm, we could describe this problem as having two hidden random variables for each team – we will call them o and d for offense and defense respectively. These variables are assumed to be normally distributed with means μ_o , μ_d and variances σ_o^2 , σ_d^2 respectively. We then assume our known variable s, representing the actual score, to be normally distributed, with its distribution given by $\eta((o+d)/2,\sigma^2)$ for some fixed σ^2 . When we look at a training example, which in this problem is a game result, then we can designate one team t_r as the "reviewer," and say that the other team t_s is the subject of the review. Then, we can calculate the intrinsic values of $O(t_s)$ and $D(t_s)$ by using the game score as a review of these values by t_r , while considering the biases of t_r to be its intrinsic values $O(t_r)$ and $D(t_r)$. To do so, we require two parallel sets of E and M steps which update $O(t_s)$ and $D(t_s)$, respectively. In interest of space, we will only look at one of these steps while understanding that a similar parallel step exists.

The steps for the algorithm proceed as follows (note the subscipt numbers indicate the value of a variable for a particular team (1 or 2) in a game):

```
\begin{split} &(\text{E-step}) \\ &\text{For each } i, \text{ set } Q_i(o_1^{(i)}, d_2^{(i)}) := p(o_1^{(i)}, d_2^{(i)} \mid s_1^{(i)} \; ; \; \mu_{o1}, \; \mu_{d2}, \; \sigma_{o1}^{2}, \; \sigma_{d2}^{2}) \\ &= p(s_1^{(i)} \mid o_1^{(i)}, d_2^{(i)} \; ; \; \sigma^2) * p(o_1^{(i)} \; ; \; \mu_{o1}, \; \sigma_{o1}^{2}) * p(d_2^{(i)} \; ; \; \mu_{d2}, \; \sigma_{d2}^{2}) / p(s_1^{(i)}) \\ &= w_i(o_1, d_2) \end{split} &(\text{M-step}) \\ &\text{set } \Theta := \text{argmax}_{\Theta} \sum_i \sum_{o(i)} \sum_{d(i)} w_i(o_1, d_2) * \log((p(s_1^{(i)}, o_1^{(i)}, d_2^{(i)}; \Theta) / w_i(o_1, d_2)) \\ &= \text{argmax}_{\Theta} \sum_i \sum_{o(i)} \sum_{d(i)} w_i(o_1, d_2) * \log((p(s_1^{(i)} \mid o_1^{(i)}, d_2^{(i)}) * p(o_1^{(i)}; \Theta) * p(d_2^{(i)}; \Theta) / w_i(o_1, d_2)) \\ &\quad \text{, where } \Theta \text{ is our four parameters } \mu_{o1}, \; \sigma_{o1}^{2}, \; \mu_{d2}, \; \sigma_{d2}^{2} \end{split}
```

In the M-step lies the tractability issue. If we solve the gradient of the M-step equation for any of the parameters, we will find that we must perform 2 sums to set the parameter – one over all i's and one over all values of $o^{(i)}$ or $d^{(i)}$. In football, it is reasonable for either $o^{(i)}$ or $d^{(i)}$ to take any of a range of 80 or so values. This range of values is too large to continually calculate a probability distribution over, and will cause a very large increase in execution time when compared with the paper/review problem which served as the inspiration for this idea. In the paper/review system, the domain was completely arbitrary and could be easily shrunk to allow easier computation.

At this point in the process, I was faced with a design decision. Either I could attempt to compress the domain of possible scores, or I could simplify the algorithm to allow the full domain of scores. I chose the latter, as one of the original motivations behind this problem was that I felt that current systems used for the BCS are in error when their rankings are determined only by the win/loss outcome of the game and not by the actual score.

To simplify the EM-based algorithm while keeping the motivation for the original idea, I kept the original ideas of "intrinsic" offense and defensive ratings for each team, but changed them such that they are not a probability distribution. However, they still represent the same ideas, which is that team i "reviews" team j's offense using the actual points scored by j against i and the bias of i is represented by its intrinsic defensive rating. Of course, the opposite is also true if you replace "defense and offense" in the previous sentence. Applying these principles led to the following algorithm.

V. The Algorithm

For each game between two teams i and j, we would like to set the offensive rating of i and the defensive rating of j such that $((o_i + d_j)/2)$ - $s_{ij} = 0$. However, this will surely not be possible across all games, so we will measure our performance by taking the sum of the squared error for each game:

$$\sum_{i}\sum_{j} ((o_i + d_j)/2) - s_{ij})^2$$

This equation can be compared to the M-step equation of the EM algorithm, with the intention that we will update the parameters o_i and d_i for each i, j with the intention of

maximizing the term above. We can do this by taking the derivative of the equation above w.r.t o_i and d_i to yield the update rule:

$$\begin{aligned} o_i &:= \left(\sum_j 2^* s_{ij} + d_j\right) / \, n \\ d_i &:= \left(\sum_j 2^* s_{ij} + o_j\right) / \, n, \text{ where n in both cases } \text{ is the number of teams on i's schedule.} \end{aligned}$$

These update rules were computed after each game, for each team, until convergence. An example of generated rankings is included in the appendix. These rankings are generated by playing a round robin between all teams, and using the overall winning percentage in this round robin to rank the teams.

VI. Conclusions/Results

As can be seen in the chart included in the appendix, this algorithm produced very positive results which are competitive with the best prediction algorithms in this domain. The chart below includes results from the last 5 years, and shows that all years' results trend upwards with the number of games played, with 4 of the 5 years trending towards 75% correct by year's end, and the 5th trending just below 70% at the end of the season. This points to the fact that the program could have benefited from more training examples, and was still learning at the end of the season. While at first glance, the consistent drop in accuracy for the final 50 games of each season seems strange, this can safely be attributed to the fact that the end of the season consists of conference championships and bowl games which are set up to be competitive. Thus, the last 50 or so games of each season will always be more difficult to predict than the rest for human and computer alike. The fact that we still correctly guess these games with 65-70% accuracy points to the fact that this algorithm can work even when faced with difficult decisions.

The main disadvantage to the simplification from the EM algorithm is the inability to create something equivalent to the E-step, which places a "weight" on training examples. The update equation I used could be modified to use a multiplicative weight in front of the error term, but this weight could not be learned (as it would always be zero). In attempts to set arbitrary weights which I thought should work based on my knowledge of football, the unweighted algorithm always produced more accurate results. I believe this may come from the fact that the training set used in calculating each parameter is very small, and weighting examples further decreases the diversity of examples which lead to the ratings.

Also rejected quite conclusively by this algorithm were attempts to "filter" the score going into the algorithm. For example, I had assumed before beginning that using a logarithmic curve to modify actual scores would create a more accurate representation of team's abilities, since it would reduce the effects of victory by a very large margin. I tried applying a logarithmic filter and this produced undesirable results. This actually makes sense, as the algorithm accounts for a team's propensity to give up large amounts of points and thus does not overly reward large margins of victory.

My further work in the college football ranking domain will consist of taking the opposite path in the design process, and using a more complex algorithm with simplified data. This will provide an interesting case study once the other method is implemented and the quality of the results are compared.

VII. Appendix

Current Rankings – December 16, 2005 (for result analysis, see attached file)

UPF=Unbiased Points For (called o in this paper), UPF=Unbiased Points Against (d)

The number before a team's name is its round robin winning percentage.

1) 1.0 Texas UPF=83.14 UPA=-2.83 2) 0.992 Southern California UPF=83.07 UPA=-0.02 3) 0.983 Ohio State UPF=51.62 UPA=-16.86 4) 0.975 Penn State UPF=56.33 UPA=-8.42 5) 0.966 Virginia Tech UPF=52.76 UPA=-3.35 6) 0.958 Notre Dame UPF=63.04 UPA=9.17 7) 0.949 Michigan UPF=45.79 UPA=-7.58 8) 0.941 Auburn UPF=52.62 UPA=3.21 9) 0.932 Louisville UPF=69.21 UPA=21.35 10) 0.924 Oregon UPF=55.15 UPA=7.63 11) 0.915 Miami (Florida) UPF=41.21 UPA=-5.57 12) 0.907 Iowa UPF=42.46 UPA=-4.11 13) 0.898 Texas Tech UPF=50.71 UPA=7.39 14) 0.89 Minnesota UPF=61.4 UPA=18.15 15) 0.881 West Virginia UPF=44.37 UPA=1.86 16) 0.873 Georgia UPF=39.84 UPA=-1.89 17) 0.864 Louisiana State UPF=43.2 UPA=4.21 18) 0.856 Wisconsin UPF=50.45 UPA=12.98 19) 0.847 Arizona State UPF=53.55 UPA=16.55 20) 0.839 California UPF=42.64 UPA=5.98 21) 0.831 Michigan State UPF=52.1 UPA=15.75 22) 0.822 Boston College UPF=32.49 UPA=-1.39 23) 0.814 Florida UPF=40.83 UPA=7.57 24) 0.805 Alabama UPF=29.22 UPA=-3.78 25) 0.797 Purdue UPF=45.08 UPA=12.94 26) 0.788 Iowa State UPF=37.78 UPA=5.75 27) 0.78 Oklahoma UPF=38.28 UPA=7.29 28) 0.771 Texas Christian UPF=40.38 UPA=10.0 29) 0.763 UCLA UPF=59.17 UPA=28.92 30) 0.754 Northwestern UPF=52.88 UPA=22.95 31) 0.746 South Florida UPF=33.14 UPA=4.57 32) 0.737 Clemson UPF=35.78 UPA=7.23 33) 0.729 Fresno State UPF=42.4 UPA=14.96 34) 0.72 Florida State UPF=40.21 UPA=13.49 35) 0.712 Georgia Tech UPF=26.94 UPA=2.92 36) 0.703 Colorado UPF=31.78 UPA=8.93 37) 0.695 Texas A&M UPF=48.13 UPA=25.89 38) 0.686 Nebraska UPF=31.92 UPA=10.1 39) 0.678 Stanford UPF=34.82 UPA=13.81 40) 0.669 Arizona UPF=29.16 UPA=9.36 41) 0.661 Washington State UPF=43.62 UPA=25.19 42) 0.653 Tennessee UPF=22.14 UPA=4.1 43) 0.644 Boise State UPF=44.17 UPA=26.57 44) 0.636 South Carolina UPF=30.51 UPA=13.54 45) 0.627 Arkansas UPF=32.88 UPA=16.73 46) 0.619 Virginia UPF=33.61 UPA=17.55 47) 0.61 Maryland UPF=33.85 UPA=18.11 48) 0.602 Tulsa UPF=36.7 UPA=21.69 49) 0.593 North Carolina State UPF=23.29 UPA=9.08 50) 0.585 Pittsburgh UPF=27.72 UPA=14.98 51) 0.576 Brigham Young UPF=41.88 UPA=29.84 52) 0.568 Northern Illinois UPF=34.61 UPA=24.07 53) 0.559 Missouri UPF=39.66 UPA=30.35

54) 0.551 Southern Mississippi UPF=31.55 UPA=22.84

55) 0.542 Connecticut UPF=23.77 UPA=15.07

56) 0.534 Oregon State UPF=33.35 UPA=24.71

59) 0.508 North Carolina UPF=23.18 UPA=15.11

57) 0.525 Kansas UPF=21.3 UPA=12.73 58) 0.517 Washington UPF=23.44 UPA=15.11

60) 0.5 Rutgers UPF=31.73 UPA=23.74

```
61) 0.492 Miami (Ohio) UPF=38.14 UPA=30.49
62) 0.483 Utah UPF=32.12 UPA=24.78
63) 0.475 Wake Forest UPF=33.9 UPA=26.68
64) 0.466 Indiana UPF=30.16 UPA=24.52
65) 0.458 San Diego State UPF=27.04 UPA=22.22
66) 0.449 Navy UPF=33.12 UPA=28.31
67) 0.441 Baylor UPF=20.85 UPA=17.18
68) 0.432 Vanderbilt UPF=37.2 UPA=33.68
69) 0.424 Kansas State UPF=31.05 UPA=27.56
70) 0.415 Houston UPF=31.68 UPA=28.89
71) 0.407 Toledo UPF=29.57 UPA=28.16
72) 0.398 Colorado State UPF=26.58 UPA=25.39
73) 0.39 Memphis UPF=21.15 UPA=21.67
74) 0.381 Texas-El Paso UPF=32.79 UPA=33.35
75) 0.373 Air Force UPF=34.2 UPA=36.26
76) 0.364 Central Michigan UPF=18.67 UPA=20.77
77) 0.356 Alabama-Birmingham UPF=26.48 UPA=28.76
78) 0.347 New Mexico UPF=29.17 UPA=31.67
79) 0.339 Bowling Green State UPF=32.6 UPA=35.1
80) 0.331 Central Florida UPF=26.52 UPA=30.58
81) 0.322 Wyoming UPF=21.9 UPA=27.09
82) 0.314 Louisiana Tech UPF=23.24 UPA=30.16
83) 0.305 Hawaii UPF=32.34 UPA=39.8
84) 0.297 Nevada UPF=30.11 UPA=38.65
85) 0.288 Syracuse UPF=10.0 UPA=19.26
86) 0.28 Oklahoma State UPF=22.5 UPA=32.26
87) 0.271 Kentucky UPF=25.28 UPA=35.34
88) 0.263 Mississippi UPF=8.08 UPA=18.24
89) 0.254 Army UPF=16.85 UPA=27.31
90) 0.246 Southern Methodist UPF=13.24 UPA=23.96
91) 0.237 Mississippi State UPF=7.99 UPA=18.92
92) 0.229 Akron UPF=15.71 UPA=26.79
93) 0.22 East Carolina UPF=21.76 UPA=32.88
94) 0.212 Cincinnati UPF=21.5 UPA=33.73
95) 0.203 Illinois UPF=21.99 UPA=37.03
96) 0.195 Western Michigan UPF=30.39 UPA=46.07
97) 0.186 Marshall UPF=7.62 UPA=25.16
98) 0.178 Eastern Michigan UPF=12.34 UPA=32.1
99) 0.169 Middle Tennessee State UPF=2.57 UPA=27.43
100) 0.161 Ball State UPF=19.93 UPA=47.55
101) 0.153 Tulane UPF=13.14 UPA=41.59
102) 0.144 Ohio UPF=7.53 UPA=36.01
103) 0.136 Rice UPF=16.9 UPA=46.93
104) 0.127 Duke UPF=13.88 UPA=44.55
105) 0.119 Louisiana-Lafayette UPF=10.4 UPA=43.3
106) 0.11 Nevada-Las Vegas UPF=6.72 UPA=42.05
107) 0.102 San Jose State UPF=7.26 UPA=42.62
108) 0.093 Utah State UPF=5.26 UPA=41.74
109) 0.085 Arkansas State UPF=3.94 UPA=44.75
110) 0.076 Kent UPF=-1.93 UPA=39.16
111) 0.068 Idaho UPF=10.85 UPA=52.46
112) 0.059 Louisiana-Monroe UPF=8.97 UPA=53.09
113) 0.051 Troy State UPF=-9.24 UPA=37.67
114) 0.042 Florida International UPF=6.34 UPA=53.46
115) 0.034 New Mexico State UPF=2.43 UPA=49.83
116) 0.025 Buffalo UPF=-10.18 UPA=38.56
117) 0.017 Florida Atlantic UPF=-8.87 UPA=44.3
118) 0.0080 Temple UPF=-1.91 UPA=53.07
119) 0.0 North Texas UPF=-6.69 UPA=49.43
```