Bowl Championship Series Vulnerability Analysis

Caio Soares
Dept. of Computer Science and Software Engineering
Auburn University, AL 36849
(334) 844-6333
soarecv@auburn.edu

George Page
Dept. of Computer Science and Software Engineering
Auburn University, AL 36849
(334) 703-0117
pagegeo@auburn.edu

Jonathan MacDonald
Dept. of Computer Science and Software Engineering
Auburn University, AL 36849
(334) 844-7002
macdojr@auburn.edu

Sanjeev Baskiyar
Dept. of Computer Science and Software Engineering
Auburn University, AL 36849
(334) 844-6306
baskiasa@auburn.edu

John Hamilton, Jr.
Dept. of Computer Science and Software Engineering
Auburn University, AL 36849
(334) 844-6360
hamilj1@auburn.edu

Gerry Dozier
Dept. of Computer Science
North Carolina A&T State University
Greensboro, NC 27411
(336) 334-7245 ext. 467
gvdozier@ncat.edu

ABSTRACT

The Bowl Championship Series (BCS) is the system used to select the top two college football teams to play in a championship game at the end of a season. During the nine seasons played under the BCS, few have not caused considerable debate and controversy. We believe that an analysis of the BCS and its components is therefore required. This work uses a GRNN to model the eight polls that compose the ranking system of the BCS. Then, a genetic algorithm (GA) is used to evolve season schedules that would perform poorly under the current BCS. These difficult seasons are then analyzed to highlight weaknesses in the current model.

Keywords

Bowl Championship Series (BCS), General Regression Neural Network (GRNN), Genetic Algorithm (GA).

1. INTRODUCTION

The BCS was introduced following the 1998 college football season as a system of pairing the two best Division 1A teams in a National Championship Game. The BCS formula is a compilation of human opinion polls and computer ranking polls. The formula currently consists of an equally weighted average of the Harris Interactive Poll, the USA Today Coach's Poll, and a computer ranking. The computer ranking is itself a result of 6 ranking algorithms: Anderson & Hester, Billingsley, Colley Matrix, Massey, Sagarin, and Wolfe; in which the highest and lowest rankings are removed and the other four are averaged. The formula is often revised following seasons in which there is controversy about the teams chosen to play in the championship game, usually without any transparent explanation. In this paper, a vulnerability analysis of the current formula will be performed.

A GRNN is utilized to model all eight polls. The polls need to be modeled because each computer poll's algorithm is proprietary and often not public [5]. Then, a GA is used to evolve season schedules that are difficult for the current BCS to evaluate [1]. These artificial seasons are then used to analyze weaknesses in the BCS formula.

2. DATA COLLECTION

2.1 Poll Data

In college football, it is traditional for only the top 25 teams to be ranked by the polls. Thus, for each week of the 2005-2006 and 2006-2007 college football seasons 25 entries from each poll are extracted [6]. Only the last two years are used because the BCS formula was revised following the 2004-2005 season. When necessary, the rankings are normalized with 1 being the highest possible ranking and 0 being the lowest. To determine the predictive ability of the ranking, data is also collected as to how well each given

Richard Tapia Celebration of Diversity in Computing Conference'07
October 14–17, 2007, Orlando, Florida, USA.

team performed in their game the subsequent week. A team is given a result of 1 if they either won their game or lost to a higher ranked team. A team is given a 0 if they either lost to a lower or unranked team, or were dropped from the rankings. In total, for the 2005-2006 and 2006-2007 seasons, 350 total instances are extracted.

2.2 Historical Data

The cumulative winning averages for each Division 1A football team are also collected from the last six years [4]. This is used for the purposes of simulating seasons, based on the idea that teams will continue to win at a similar rate in the future.

3. POLL MODELING

A separate GRNN is used to model each poll [5]. Each GRNN is trained based on the historical BCS data collected in the data collection phase.

3.1 Ranking Format

The ranking format utilized by our training algorithms consists of a given week's ranking and result data as inputs and the next week's ranking as desired output.

3.2 K Values

Since teams do not play in a vacuum, polls will take into account how well other teams do when evaluating a new ranking for a given team. To simulate this in a learning algorithm, a K value is used to increase the vector size to include the (K-1) / 2 neighbors above and below a team of a given rank. Thus, finding an optimal K will give us the best balance between a learner that does not generalize well and one that will overfit.

3.3 GRNN

A GRNN is trained and tested based on data collected from each poll, utilizing a leave-one out training scheme for the validation set, which produces an average error that is used to represent the generalizing ability of the GRNN.

To find an optimal sigma for the GRNN, we utilize a Standard Evolutionary Program (SEP) from the XTOOLS software suite [8]. This program looks for an optimal sigma between 0 – 100 using a population of 20 and a maximum of 500 function evaluations.

3.4 Poll Modeling Results

As a result of testing various K values, the lowest error is found with K=3. This K value, along with a set of optimal sigmas for each poll, is saved for use in evaluating seasons created by the GA.

4. EVOLVING SEASONS

Since there are only a finite number of seasons available to test potential improvements to the BCS, a GA is used to evolve a set of seasons found to be difficult for the current BCS formula. In order to evolve a set of difficult seasons, first seasons have to be randomly generated. Next, these seasons are passed to a GA to be mutated and evolved. For the purposes of the GA, the difficulty of a season is considered to be the difference between the computed rankings of the second and third teams. The GA will then attempt to minimize this difference as its fitness function. This divide is chosen because the BCS currently chooses the top two teams to play in the national championship game.

4.1 GA

The GA uses a generational replacement strategy with mutation for diversity and an elite individual surviving between generations in order to not lose the very best solution found. Each individual in the GA is an entire season's schedule of games. A generation of μ individual schedules is initially generated randomly, and then evolved by mutation based on the previous generation's season's best fit $\mu/2$ individuals, with the best parent kept between generations. Each season is mutated using a mutation rate computed by selecting a base mutation rate then multiplying that times a random Gaussian number with 0 mean and 1 standard deviation. Then, that is normalized to a percentage of the total number of weeks, becoming the number of weeks removed from the end of the season. Finally, a new end-of-season is randomly generated starting from the last week kept. The following process determines fitness. Each game of the season is played on a

week-by-week basis. After each week, the results of the games are input to the modeled polls, and new BCS rankings are produced. After the final week of each season, the separation between the second and third ranked teams in the final BCS formula is recorded as that individual's fitness.

4.1.1 Game Prediction

For each game, a result forecast is calculated to predict which team wins or loses. This calculation is based on the Bill James Log5 method [7] for estimating a team's chance of winning over another team using each team's winning percentages. The individual team's winning percentage is calculated from a percentage of its historical data gathered in Section 2.2, and a proportion of its winning percentage of this season.

4.1.2 Weekly Polls

Following each week a new set of poll values is calculated using the eight GRNN's trained in section 3. The input vector for each GRNN consists of the poll ranking of each team in last week's top 25, the rankings of it's K - 1 neighbors and the result for each team using the same method as described in Section 2.1. The input vector for each team is given to the GRNN, which returns the new ranking for that team. If that ranking is below a certain threshold, the team is dropped from the poll and a new team is added in its spot. The new ranked team will be the team with the highest winning percentage that is not ranked in the previous week's poll. The threshold used for dropping a team is 0.04.

4.1.3 Fitness Evaluations

After every week of a season is played and the final BCS ranking is calculated, fitness is evaluated by subtracting the ranking value associated with the 3rd team from the ranking value of the 2nd. This divide is chosen based on the idea that the less difference between these teams, the more controversy there would be about which team comes out on top.

After each generation, the child with the best fitness is compared to the elite parent, and if it's lower, the elite individual is replaced. The mu/2 individuals with the best fitness are then used as the parents for the next generation. The final output of the GA is a single season's schedule, the results of the games played, and the weekly poll rankings.

5. VULNERABILITY ANALYSIS

5.1 Methodology

In the current BCS formula, the highest and lowest computer polls are dropped from the rankings and the remaining four are averaged. Therefore, data on how often each computer is dropped is collected to determine if there is significant difference in utility between polls. The way prediction and contribution are computed is by multiplying a team's ranking (1 – 25, or 0 if not ranked) times the team's game result for that week (0 or 1) as computed in Section 2.1.

This comparison will give insight as to the predictive ability of each poll, individually and in comparison to the BCS. In addition, there lies much greater importance in correctly predicting higher ranked teams than lower ranked teams, given that every season the BCS is trying to correctly pick the top two teams for a National Championship. So, for a given week, a sum of points is accumulated for each poll.

In the first method, poll prediction, each poll uses the rankings of its own 25 top ranked teams and their corresponding 25 game results based on its own poll rankings. This method shows the predictive ability of each poll on its own, without influence from, or on, the BCS. In the second method, BCS prediction, each poll uses the rankings from its own 25 top ranked teams, but the game results used are the ones calculated by the BCS rankings. This method shows how the BCS predicts the teams picked by each individual poll. In the final method, BCS contribution, each poll uses the rankings from the 25 top ranked teams of the BCS, but the game results from its own poll. The final method shows how well each poll can predict the teams picked by the BCS.

5.2 Vulnerability Results

Analysis is based on 20 difficult seasons generated by the GA. The first measure of competition between polls is the percentage of times each computer poll is dropped when calculating the BCS. This study found that there is not much difference in the percentage of times each poll is dropped. This suggests that all polls take turn in predicting too high or too low, and do so at a fairly even rate. The Massey Poll is found to be the most dropped poll at roughly 20% of the time and Wolfe the least dropped poll, at roughly 14% of the time.

The more interesting results arise from the three contribution methods discussed in 5.1. Under all three methods, each of the 8 polls fall into a very close range, contributing between 65-70% of the time. This seems uninteresting until you consider that the two human polls are weighted much more heavily than the computer polls that they do not significantly out-perform.

6. CONCLUSION

This analysis of the current BCS shows a significant vulnerability in the weights used in its formula. The current BCS formula weighs the human polls 1/3 each, and an average of four median computer polls the other 1/3. This allows the human polls to unfairly influence the BCS results, while the computer polls have a much smaller impact. These results suggest that all of the polls, human or computer, are roughly the same when it comes to correctly ranking teams.

It seems inevitable that they BCS formula will be revised again in the future. This study has found a significant concern with the way in which polls are currently weighted, which can hopefully be addressed when such a revision happens. As future work, the authors hope to develop new potential formulae based on the kind of analysis performed in this study.

7. ACKNOWLEDGMENTS

Thanks to our classmates: Taha Brahim, Haiquan Chen, Jason Davis, Brent Harrison, Idongesit Mkpong-Ruffin, Brandi Smith, and Lei Zhang.

8. REFERENCES

[1] Back, T., U. Hammel, and H. Schwefel. "Evolutionary Computation: Comments on the History and Current State." *IEEE Transactions On Evolutionary Computation* 1, no. 1 (1997).

[2] Dozier, G., A. Homaifar, E. Tunstel, and D Battle. "An Introduction to Evolutionary Computation." In *Intelligent Control Systems Using Soft Computing Methodologies*, by A. Zilouchian & M. Jamshidi (Eds.), 365 - 380. CRC Press, 2001.

[3] Hogg, R., and E. Tannis. "Tests of Statistical Hypotheses (Chapter 8)." In *Probability and Statistical Inference*, 495. Prentice Hall, 2001.

[4] National Collegiate Athletic Association. *Statistics - NCAA Sports.* 2007. http://www.ncaasports.com/football/mens/stats/stats (accessed Apr 26, 2007).

[5] Specht, D. "A General Regression Neural Network." *IEEE Transactions on Neural Networks*, 1991.

[6] The National Football Foundation & College Hall of Fame, Inc. *BCS Standings Archive.* Jan 2007. http://www.bcsfootball.org/bcsfb/archiveStandings (accessed Apr 26, 2007).

[7] Tippett, Tom. *May the Best Team Win ... at Least Some of the Time.* Oct. 2002. http://www.diamond-mind.com/articles/playoff2002.htm (accessed Apr. 2007, 2007).

[8]XTOOLS Software Suite. http://www.eng.auburn.edu/~gvdozier/xtoools