CS 224W Project Final Report:
Predicting Super Bowl Winners Through Graph Analysis

Victoria Kwong
vkwong@stanford.edu

Tuesday, December 9, 2014

Introduction

Super Bowl XLVIII drew the largest television audience in American history with 111.5 million US viewers.[1] This surpassed the previous record set by Super Bowl XLVI two years prior with 111.3 million viewers.[2] In fact, the last four Super Bowls have been the four most watched TV programs in U.S. history.[20] Despite the uneventful game between the Denver Broncos and the Seattle Seahawks, where the Seahawks defeated the Broncos 43 to 8, the power of the Super Bowl drew in a record-shattering audience, highlighting the pull of this live event.[3] With the population of the United States at 316.1 million, the Nielsen ratings suggest that more than a third of all Americans were watching Super Bowl XLVIII.[4]

This obsession with the Super Bowl is similarly translated to the Vegas sportsbooks, where gamblers bet a record $119.4 million in the Nevada casinos alone.[5] This project seeks to identify the Super Bowl winner for a given football season. If successful, the project could have massive ramifications on Vegas sportbooks since the Super Bowl is the most gambled-on sports event in the United States. Vegas has accurately predicted the winner of the Super Bowl for 68.75% of the last 48 Super Bowls, demonstrating the difficulty of predicting this event[6] Thus any algorithm that comes close to this percentage could significantly alter the earnings of the Vegas casinos, which netted $19.7 million this year on this one game alone.[7]

The goal of this paper is to frame the NFL season as a graph. Then utilizing different graph properties and algorithms such as the PageRank algorithm, we will try to model the actual ranks of each of the football teams. After we have computed the final values, we can then determine who the victor of the Super Bowl will be.

Relevant Prior Work

Article 1: Who Is the Best Player Ever? A Complex Network Analysis of the History of Professional Tennis[8]

This article seeks to uncover the best tennis players between the years 1968 and 2010. To do so, [8] uses a dataset of all matches in Grand Slams and ATP World Tour tournaments played by professional tennis players. This data is then translated into a directed and weighted network graph where the nodes represent the professional players and a directed edge from node i to node j represents that node j has defeated node i in a match. Additionally, [8] uses weight w_{ij} for edge from i to j to represent the number of times node j has defeated node i. It then uses a diffusion algorithm to calculate a "prestige score" to determine who is the best tennis player during that forty-two year period.

Essentially the method described above is the PageRank algorithm utilized for tennis players. As such, it is the tennis equivalent of what we are trying to achieve with the NFL. It presents a comparison on how the PageRank algorithm works with different sports. Moreover, it provides a guideline on how to model our PageRank algorithm to adjust for varying levels of strength.

Article 2: Understanding baseball team standings and streaks[9]

[9] focuses on the rankings of the baseball teams and models each team as a node with the edge between node i and node j reflecting the probability that team i will beat team j. It assumes that each team has an intrinsic strength x_i and with the Bradley-Terry model the probability of team i winning is simply:

$$p_{ij} = \frac{x_i}{x_i + x_j}$$

Thus to calculate the final winning fraction of a team i, it is simply:

$$W(i) = \frac{1}{N} \sum_{j=1}^{N} \frac{x_i}{x_i + x_j}$$

where N is the number of games a team i plays. More specifically, [9] assumes that x_i is a value between 0 and 1 and there exists an x_{min} such that no team has less "strength" than x_{min}.

This concept can just as easily be applied to football teams. We would need to come up with our own "intrinsic strength" values as the paper doesn't specify how it comes up with their value. Additionally this simplistic model provides a good comparison to the PageRank algorithm.

Data

All the data for this project was downloaded from `http://www.repole.com/sun4cast/data.html`. This site provided CSV files of pre-season, regular season, and post-season data from 1978 to present day. Each file represents all of the pre-season, regular season or post-season games for a given year. Thus for each year there are three CSV files. Each line of the files represent one game. The information for each game includes the names of the two teams, the score, the date of the game, the line and the total line.

To extract the Super Bowl winners from the data, we parse the post-season data files. Grabbing the information from the last line of each of the post-season data files, we have a list of the two teams in the Super Bowls and the score for each team.

In order to model the NFL season as a graph, we then need to parse the regular season data such that it is readable by Snap.py.

Algorithms & Methods

PageRank

The core of this proposal is to utilize PageRank to model the NFL season and then given the two teams playing in the Super Bowl, predict the SuperBowl winner from the model. PageRank is an iterative method. We begin by giving all the teams the same weight. Since we want the pagerank values to sum to 1, we start by giving each team:

$$p_{i,0} = \frac{1}{|N_{teams}|}$$

Let's start with the straightforward version of the algorithm. For the entire season, we create a single directed graph where each node represents a team in the NFL. Now for every game played in the season, we draw an edge from the losing team to the winning team. In other words if the San Francisco 49ers lost to the Atlanta Falcons, there would exist an edge from the node representing the San Francisco 49ers pointing to a node representing the Atlanta Falcons. Thus for the modern NFL, the graph will consist of 32 nodes and 256 edges.

PageRank is an iterative algorithm. To find the pageranks of each of the teams, we simply repeat the following calculations until the pagerank values converge. For team i at iteration t, its pagerank is:

$$p_{i,t} = \alpha \cdot p_{i,t-1} + (1 - \alpha) \sum_{j \in E(ji)} \frac{p_{j,t-1}}{O(j)}$$

where α is a value from 0 to 1, $E(ji)$ is the set of all incoming edges to node i, and $O(j)$ is the number of outgoing edges from node j. Essentially, we are solving for the eigenvector R such that

$$R = (1 - \alpha) \begin{bmatrix} \frac{1}{N} \\ \frac{1}{N} \\ \dots \\ \frac{1}{N} \end{bmatrix} + \alpha M R$$

where M is an adjacency matrix that has a 1 if there exists a directed edge between nodes i and j and 0 otherwise. The logic behind this algorithm is that the a team derives $1 - \alpha$ of its pagerank from the teams it defeats from pure skill and α of its pagerank from "luck" and simply happening to win games.

However, using the straightforward version of the algorithm may not be the proper model for this NFL ranking problem because its lack of flexibility cannot capture the complexity and intracies of the NFL. As such, perhaps a better way to model the NFL is to calculate pagerank on a weekly basis. As such, we now construct a unique directed graph for each week in the season where each node represents a team. The edges are again constructed in the same manner where an edge points from a losing team to a winning team. Then for each week t, we can calculate the pagerank values such that

$$p_{i,t} = \alpha \cdot p_{i,t-1} + (1 - \alpha) p_{j,t-1}$$

where α is a value from 0 to 1 and j is the team that team i beat that week. This algorithm is simpler than the one above we used for the entire season because we know that for each week, there can at most be one outgoing edge

from any given node since each team only plays one game each week.

In the modern NFL, we do this for a total of 17 times for each of the 17 weeks of the regular season. After this, we have a final set of pagerank values for the teams. Ideally now the pagerank values are representative of the rankings of the teams. We then query the pagerank values for the two teams in the Super Bowl and predict the winner to be the team with the higher pagerank value.

Doing this variation of the PageRank algorithm let's us factor in the team's varying strength at different points of the season. For example, suppose team i has won all of its first 10 games and proceeds to lose the following 3. Then when team j beats team i in the 11th game, it should mean more than if team j beat team i in the 13th game because at that point two others team have already demonstrated that team i is beatable. Team i was at its peak strength during its 11th game, having beat its first 10 teams but at game 13, it's clear that this is no longer true. Perhaps games 11 and 12 caused a lot of injuries and thus leaving team i vulnerable. Whatever happened, it's obvious that team i has varying levels of strength and the weekly PageRank variation is able to capture these fluctuations.

On top of the base model, we can build some optimizations into the algorithm. For example, in the NFL, teams that are playing at home win their game 57.3% of the time, indicating that there is a slight home field advantage.[10] Using this information we can adjust α accordingly. For example, if a team loses at home, they should lose more of their pagerank value because statistics suggest that they should win the majority of the time when playing at home. As such, when a team plays at home, we can decrease α.

Still a game played later in the season tells us more about the team (with regards to its chances of winning the Super Bowl) than in the beginning of the season. Intuitively this makes sense as many things can happen over the course of the season and it's best to predict how well a team will do relative to how well it has been performing recently. To take this information into account, we scale α proportional to the week of the season. Essentially the idea is that the winning team takes less of the losing team's pagerank in the beginning of the season and more towards the end of the season as later wins are more telling about the team's strength.

Additionally, another optimization we can build into the algorithm is point differential. Teams that tend to do better have a better point differential.[11] In other words, they score more points and have less points scored on them. As such, this can be incorporated into their pagerank score. For example, if team i defeats team j by a huge margin, team i should be able to take more of team j's pagerank than if team i barely defeated team j. As such, when a team wins with a larger point differential, we decrease α.

Another optimization we considered using was preseason data. Prior to the beginning of the regular season, there are several weekends of exhibition games. However, NFL commission Roger Goodell has famously complained about the low quality of preseason games, noting that veteran players usually don't play in these games to lower the risk of injury.[12,13] As such, preseason games relay little information about the actual season.[14]. Thus, we opted to not factor in preseason data in our algorithm.

Bradley-Terry-Luce Model

As mentioned in [9], Bradley-Terry model is a simple probability model that is used to predict which team will win. The probability that team i will beat team j is

$$p_{ij} = \frac{x_i}{x_i + x_j}$$

x_i is a real value that is used to represent the skill level of the team. This proposal will explore different values of x_i to see which value will provide the most accurate prediction.

Again we model the NFL season as a graph where each node represents a team in NFL. There exists two directed edges e_{ij} and e_{ji} between team i and team j that play each other. For every game, $e_{ij} + e_{ji} = 1$ Then for any given team i, the real value for that team is simply the sum of the weights of the incoming edges:

$$x_i = \sum_{j \in V} e_{ji}$$

Let's start with the most basic version of this algorithm. Suppose team i beats team j. Then edge e_{ij} will have weight 1 and edge e_{ji} will have weight 0. As such, x_i is equivalent to the number of games that team i has won. Ranking the teams by the weighted number of wins provide an approximation guarantee of 5 for the actual ranking of the teams as long as the condition that $e_{ij} + e_{ji} = 1$ is satisfied. [15]

While the above algorithm is a good start, it fails to take into account any information about the game and teams. For a more descriptive value of x_i, let's try to factor in the

score of the game. Suppose team i and team j play each other. Now let e_{ij} and e_{ji} be the following:

$$e_{ij} = \frac{\text{team } i\text{'s score}}{\text{team } i\text{'s score} + \text{team } j\text{'s score}}$$

$$e_{ji} = \frac{\text{team } j\text{'s score}}{\text{team } i\text{'s score} + \text{team } j\text{'s score}}$$

Using this metric for the weight of the edges provides information on the strengths of the two teams relative to each other as point differentials per game provide a good sense of team rankings.[16]

Results

PageRank

Algorithm	Accuracy
PageRank for Entire Season	0.52777778
Weekly PageRank with No Optimizations	0.55555555
Weekly PageRank with HomeField	0.55555555
Weekly PageRank with Week	0.58333333
Weekly PageRank with Score	0.61111111
Weekly PageRank with All	0.63888889

The table above demonstrates the percentage of accurately predicted Super Bowl winners since 1978 by the different variations of the algorithm. We see that calculating pagerank values by modeling the entire season as a graph barely outperforms guessing randomly. Immediately evident is the fact that calculating pagerank iteratively on a weekly basis does better than modeling the season as a whole. As such, we make all of our optimizations off of this weekly pagerank model. We perform all of these calculations with the baseline $\alpha = 0.15$, selected because of its significance in the original PageRank algorithm.

We begin with our first optimization: homefield advantage. For games played at home, we decrease α to 0.1 as we expect less luck to be involved in winning a home game because as mentioned earlier, teams win 57.3% of the games they play at home. While this is a slight advantage, it is not significant enough to affect the final pagerank values. Additionally each team plays half of their games at home and no team in the Super Bowl has ever played on their home field, effectively making this a trivial advantage.[17,18]

With the next optimization, we see that incorporating the week of the season into the algorithm does increase its accuracy by almost three percent. Here we simply take α and scale it by week i / 17 because there are a total of 17 weeks in the season. This three percent increase suggests that performance later in the season is more informative about the team's final performance in the Super Bowl.

Finally, our last optimization incorporates the game's score into the algorithm. Here we first calculate how many scores separate the two teams. In other words, if the two teams are at most 8 points apart, it is still a one score game since the currently losing team can tie the game with just one touchdown and a two point conversion. We bucket the games into one score, two score and three or more score games and adjust α accordingly. Since the average team wins by 11.2 points, we will keep the two score games with the original alpha value of 0.15.[19] Now since a one score game indicates that the teams were more closely matched, we can increase α to 0.25 and similarly since three score games indicate blowouts, we can decrease α to 0.05. In other words, when the game is close, the winning team takes less of the losing team's pagerank value. This minor change increases the accuracy of our algorithm by almost 6 percent to 61.1%. This demonstrates that the point differential is a telling trait of the relative strengths of the teams.

Putting the three optimizations together, we have a final accuracy of 63.89%. While this is not as precise as Vegas' 68.75%, it comes fairly close. We see here that three optimizations can bring us within 5% of Vegas' accuracy, suggesing that with additional optimizations, we can come even closer to that number.

Now to ensure that we didn't overfit the α's to the Super Bowl data, let's run the algorithm on the remaining postseason data.

Games	Accuracy
Wild Card	0.55555556
Division Playoffs	0.61805556
Conference Championships	0.54166667
Super Bowl	0.63888889
All Post-Season Data	0.57349206

While these numbers may not be as good as the Super Bowl numbers, they all still do as well as the PageRank algorithm for the entire season suggesting that these α parameters provide a good estimation for Super Bowl winners.

Bradley-Terry

Algorithm	Accuracy
Number of Wins	0.61111111
Fraction of Score	0.69444444

[15] suggested that ordering teams by the weighted number of wins provides a good estimation for the actual ranking of teams. From the accuracy, the theory proved in [15] is reaffirmed by data. Notice how this basic calculation is already performing as well as the weekly PageRank with the score optimization, suggesting there is massive potential for this model.

However as demonstrated earlier with the PageRank algorithm, score differentials for a game are significantly more telling of the teams' strengths. When factoring in the scores of both teams, we see that this version of the Bradley-Terry model actually outperforms Vegas. Since this seems too good to be true, let's run the algorithm against other post-season data, checking the validity of this model.

Games	Accuracy
Wild Card	0.56349206
Division Playoffs	0.67361111
Conference Championships	0.68055556
Super Bowl	0.69444444
All Post-Season Data	0.64021164

The validity of the model is highlighted by both the Division Playoffs and the Conference Champsionships, where the accuracy for both types of games are in the high 60s.

PageRank vs. Bradley-Terry-Luce Analysis

Not only is the Bradley-Terry-Luce model simpler than the PageRank model, but it by far more accurate as well. While our variation of PageRank can accurately predict Super Bowl winners with 63.89% accuracy, the fractional score version of Bradley-Terry predicts the winners with 69.44% accuracy. If we look at the other post-season games with accuracies in the high 60th percentile, we can see that Bradley-Terry also performs much more consistently compared to PageRank's 55th to 61st percentiles . Thus Bradley-Terry has potential to outsmart Vegas' algorithms.

With both of these algorithms, it should be noted that they perform best with Super Bowl data. Now the Super Bowl is different from the other playoff games in one way: it is the only game that is played between the NFC and AFC. The remainder of the playoff games are played within the two conferences. This suggests that it is easier to predict wins between the two conferences than it is within conferences and that the relative strengths between the two conferences differ noticeably.

Let's take a look at the number of games won by each conference. If we look at the win differentials between the two conferences per year, we see that for most years, it is very clear that one conference dominates the other in number of wins. Interestingly enough, choosing the conference who has more wins to win the Super Bowl correlates to predicting the Super Bowl winner correctly roughly 70% of the time. Hence, we can conclude that the relative strengths of the two conferences do tend to differ significantly, which is accurately captured by the two algorithms.

Figure 1: NFC vs AFC Win Differential[21]

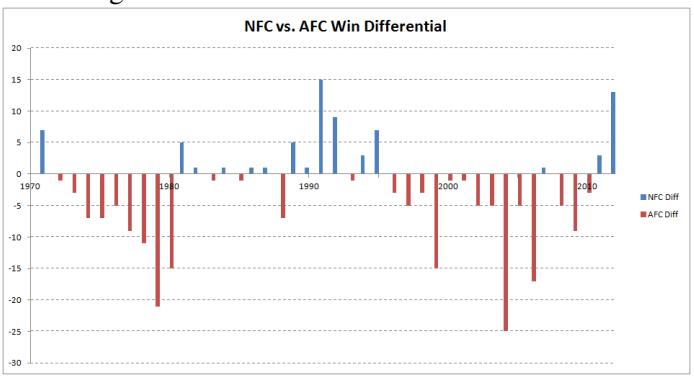
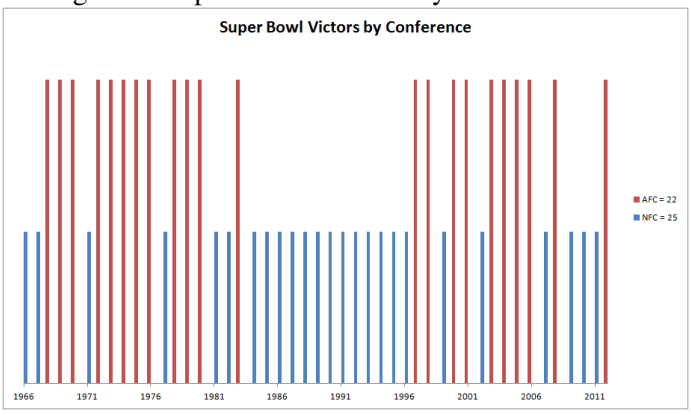


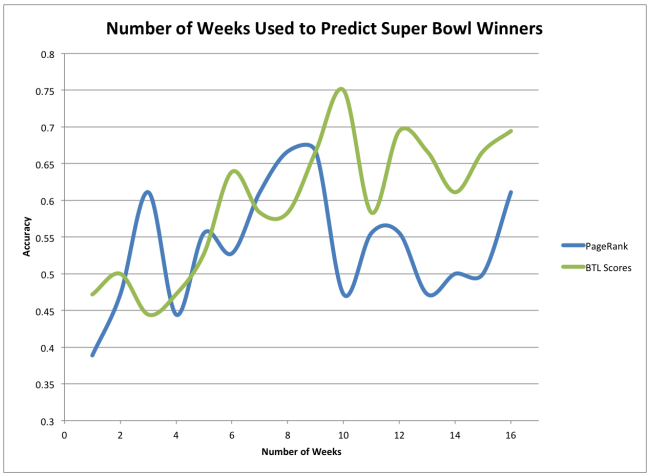
Figure 2: Super Bowl Winners By Conference[21]



Yet still, all of these numbers must be regarded with a grain of salt. There have only been 48 total Super Bowls and as such, the sample size is still small. Thus it is difficult to make generalizations off of this data alone. Perhaps it is better to use the accuracy rate derived for all post season games as there have been 396 in this same period. Now we see that the PageRank accuracy has dropped to a mere 57.35% and Bradley-Terry is at 64.02%. While neither of these are as good as Vegas' 68.75% for Super Bowls, it does have a significantly larger sample size, demonstrating that there is merit in the Bradley-Terry algorithm.

Further Investigation

One area that we explored was the number of weeks in the season needed to predict the Super Bowl winner. Using the PageRank and BTL fractional score algorithms, we have the following graph.



Here we see that there doesn't appear to be an ideal number of weeks used for PageRank accuracy and in general, it fails to follow any trend. However, for Bradley-Terry, while the accuracy does oscillate, it seems to have an upward trend in general, suggesting that the more weeks we use for our algorithm, the more accurate the final prediction will be. Again this advises us to use the Bradley-Terry model over the PageRank model for NFL predictions.

Another algorithm we considered using was the Borda count algorithm as [23] suggested that it would provide a close approximation to the optimal ranking. However, even with different variations of the count, the algorithm at most gave us a 52.78% accuracy. While this is better than randomly guessing, it came no where close to the other two algorithms we examined in our paper.

While using the fractional scoring Bradley-Terry model has yielded promising results, using the fractional scores as the intrinsic "strength" of a team may not be the best index for ranking teams. Perhaps there exists an even telling quality that we have yet to explore.

The success with the Super Bowl predictions prod us to extend this to other sporting events as well. Vegas has done notoriously terribly when it comes to predicting the winner of the World Series, such that betting on the Vegas favorite, on average, loses you 44 cents of every dollar.[22] This indicates that Vegas has severely underestimated the odds of the underdog. Similarly, with the NBA, betting on the favorite wins you 43 cents of every dollar since Vegas significantly underestimated the odds of the favorite.[22] As such, crafting a successful prediction algorithm could help one play and win in the Vegas sportsbooks.

Bibliography

[1] Taube, Aaron. "This Year's Super Bowl Was The Most Watched TV Program In U.S. History - Here's How Many People Saw It [THE BRIEF]." Business Insider. Business Insider, Inc, 04 Feb. 2014. Web. 12 Nov. 2014.

[2] Carter, Bill. "Seahawks-Broncos Super Bowl TV Ratings Top 111 Million." The New York Times. The New York Times, 03 Feb. 2014. Web. 13 Nov. 2014.

[3] Edholm, Eric. "Seahawks' Super Bowl XLVIII Victory Is the First 43-8 Final Score in NFL History." Yahoo Sports. Yahoo Sports, 2 Feb. 2014. Web. 13 Nov. 2014.

[4] "United States Census Bureau." USA QuickFacts from the US Census Bureau. United States Census Bureau, 08 July 2014. Web. 11 Nov. 2014.

[5] Associated Press. "Fans Bet Record $119M on Super Bowl." ESPN. ESPN Internet Ventures, 04 Feb. 2014. Web. 13 Nov. 2014.

[6] "Sports Betting News and Vegas Odds." VegasInsider.com. N.p., n.d. Web. 13 Nov. 2014.

[7] Chase, Chris. "Seattle's Super Bowl Win Made Gambling History." For The Win. N.p., 4 Feb. 2014. Web. 13 Nov. 2014.

[8] Radicchi F (2011) Who Is the Best Player Ever? A Complex Network Analysis of the History of Professional Tennis. PLoS ONE 6(2): e17249. doi:10.1371/journal.pone.00172

[9] Sire C, Redner S (2009) Understanding baseball team standings and streaks. Eur Phys J B 67: 473481.

[10] Dubner, Stephen J. "Home-field Advantage." NFL.com. NFL.com, 14 Dec. 11. Web. 13 Nov. 2014.

[11] Brinson, Will. "NFL Division-by-division Point Differentials for past Decade plus." CBSSports.com. N.p., 28 May 2014. Web. 05 Dec. 2014.

[12] JianaKoplos, Nancy Ammon, and Martin Shields. "Practice or Profits: Does the NFL Preseason Matter?" Journal of Sports Economics 13.4 (2012): 451-65. Department of Economics, Colorado State University, 22 June

2012. Web. 07 Dec. 2014.

[13] Vergano, Dan. "Study: NFL Preseason Games Haven't Mattered since 1994." USA Today. Gannett, 13 Aug. 2012. Web. 07 Dec. 2014.

[14] Pane, Neil. "Preseason Football Isn't Totally Meaningless (If Your Team Has A New QB)." DataLab. FiveThirtyEight, 15 Aug. 2014. Web. 07 Dec. 2014.

[15] Coppersmith, Dan, Lisa Fleischer, and Atri Rudra. "Ordering by Weighted Number of Wins Gives a Good Ranking for Weighted Tournaments." Journal ACM Transactions on Algorithms 6.3 (2010): n. pag. Ordering by Weighted Number of Wins Gives a Good Ranking for Weighted Tournaments. June 2010. Web. 08 Dec. 2014.

[16] "What Point Differential Tells You About NFL Teams - TopBet News Section." TopBet News Section. N.p., 02 Aug. 2013. Web. 08 Dec. 2014.

[17] Battista, Judy. "The Art and Science of Scheduling Meet in the N.F.L. Office." The New York Times. The New York Times, 19 Apr. 2012. Web. 09 Dec. 2014.

[18] Fabiano, Michael. "Super Bowl Curse: The Numbers Don't Lie." NFL.com. N.p., 03 Aug. 2013. Web. 09 Dec. 2014.

[19] Gordon, Aaron. "Breaking down an Average NFL Game." SportsonEarth.com. N.p., 10 Dec. 2013. Web. 08 Dec. 2014.

[20] Koba, Mark. "Super Bowl TV Ratings: Fast Facts at a Glance." CNBC. N.p., 28 Jan. 2014. Web. 09 Dec. 2014.

[21] Zhu, Mo. "Is There a Pattern of Dominance Between the NFC and the AFC?" Medium. N.p., 16 Oct. 2013. Web. 09 Dec. 2014.

[22] "Why Has Baseball Been So Hard to Predict?" PunditTracker. N.p., 1 Nov. 2012. Web. 9 Dec. 2014.

[23] Rajkumar, Arun, and Shivani Agarwal. "A Statistical Convergence Perspective of Algorithms for Rank Aggregation from Pairwise Data." Journal of Machine Learning Research 32 (2014): n. pag. Web. 19 Nov. 2014.

References

Vlahos, James. "The Super Bowl of Sports Gambling." The New York Times. The New York Times, 01 Feb. 2014. Web. 09 Dec. 2014.

Wauthier, Fabian L., Michael I. Jordan, and Nebojsa Jojic. "Efficient Ranking from Pairwise Comparisons." Journal of Machine Learning Research 28.3 (2013): 109-17. Web. 19 Nov. 2014.