Hierarchical Models for Extracting Individual Ratings from Group Competitions

Joshua Menke, C. Shane Reese, and Tony Martinez

May 26, 2006

Abstract

Providing direct and indirect contributions of more than $18 billion to the nation's gross output in 2004, the computer and video gaming industry is one of the fastest-growing sectors of entertainment. A large part of that market includes team-oriented online games. Players in these games often have a high-level of interest in statistics that help them assess their ability compared to other players, however few models exist that estimate individual player ratings from team competitions. There are models that can be used at the team-level, however the dynamic nature of the teams in the more popular public-style play of these games makes it necessary to model build team strengths from player abilities. The following presents models that describe team abilities in terms of how well the individual players on the teams contribute to their team's winning. In addition, the models presented include parameters that estimate other characteristics of the games themselves. The models are posed in a hierarchical Bayesian framework. In addition to giving players a better estimate of their skill, these models can also be used to improve current gameplay, and create more enjoyable games in the future. Companies and servers that apply well-developed statistics for assessing their players' abilities are more likely to attract and retain players, leading to greater success in the industry.

1 Introduction

According to one marketing research firm, the computer gaming industry is the fastest-growing sector of entertainment (DFC Intelligence, 2006). Robert Crandall, a senior fellow with the Brookings Institution, and Gregory Sidak, visiting professor of Law at Georgetown University stated in a recent study that "The entertainment software business provided direct and indirect contributions of more than $18 billion to the nation's gross output in 2004" (Crandall and Sidakk, 2006). In fact, "video games represent an $11 billion U.S. entertainment industry, larger than Hollywood box office sales" (DFC Intelligence, 2006). A large portion of this revenue comes from the online gaming sector, which is projected to bring in $2.5 billion in 2006. (Jarret, 2003) 44% of all video and computer gaming players say they play online in addition to offline games. (The Entertainment Software Association, 2006) One of the more popular genres in online gaming involves team competitions. This includes games based on popular sports, but more popular are team-based first-person shooter games which include blockbuster titles like Halo 2 and Halflife 2. Millions of "gamers" can be found playing these games every day.

Most online players prefer having a method of tracking their progress in the games they play, and comparing themselves to other players. If a player feels like he or she can achieve some goal with a given statistic, they are more likely to continue playing. Games and servers that provide better developed statistical methods for comparing a player's skill are more likely to retain a larger player base, and hence be more profitable. Therefore, having good ways of rating players benefits both the players, the companies producing the games, and the companies running servers for the games. In addition, if aspects of the game besides the players can be judged statistically, the developers can use this information to improve the quality of newer games. For example, in this paper we show how the fairness of the "field" two teams play on can be analyzed. This can be used to improve the fairness of gameplay.

Most players of online team games play on "public servers". By public we mean that players are free to join, leave, and switch teams as often as they like. Every match is highly dynamic in terms of which players are on either team and therefore the ratings of the teams can not be estimated at the group level, but need to be built from individual ratings. Unfortunately, for team-based games, there are few systems in place that estimate individual player ratings from group competitions. The following presents an extension of the model proposed by Bradley and Terry (1952) for paired-comparisons that estimates individual player ratings based on the winners of team competitions.

There are several types of statistics that can be tracked in these team competitive online games, many of which are interesting at the individual level, but are not consistent indicators of a player's team contribution. The goal of the models in this paper is to estimate how well a player contributes to his or her team winning the match. Therefore, the ratings are estimated only in terms of the wins a player achieves while playing on their given teams. It should be noted that the ratings here are not necessarily strong indicators of how these players would play one-on-one, but rather in team settings.

There are several reasons why ranking and rating players in public, online, team-based competitions can lead to an increase in the number of players that play and continue to play a given game or play on a given server:

- It gives players a measure of their performance and hence motivation to improve themselves. Our experience with running several online servers has shown that players tend to play more often on servers that provide a method of assessing their performance. The servers providing more statistical information are consistently full whereas the others are often near empty.

- It allows server administrators to judge the fairness of gameplay on their servers. Servers known for fair gameplay always attract more players than other servers. This is because the newer players know they have just as much a chance of winning on either team, and older players know the challenge level will remain consistent.

- It can aid players in choosing servers that better fit their abilities. If the easiness of a server can also be judged, then players can choose the servers that best fit their abilities. The following paper provides a model for comparing the difficulty level of a given server to other servers.

In the end, the games and servers that provide better statistical measures in addition to more challenging and fair gameplay will attract more players.

The game chosen for this paper is called Wolfenstein: Enemy Territory, also known as ET. It is one of the most popular team-based online games played with several hundred thousand unique players playing every day. Its popularity is partly due to the fact that it was made free by its publisher, Activision. In Enemy Territory, each match consists of an Axis side and an Allies side competing on a World War II historically inspired map. Each team has several objectives they need to complete within a certain time limit in order to win. Usually, one team team's objective is to accomplish a given goal, while the other team's objective is to prevent them from accomplishing this goal during the time limit. This often unbalanced play-style, combined with the common coming and going of players on a public server, creates matches that require rich models in order to predict. They can be compared to a soccer game where either team can have as many players as they like so long as the combined number of players on the field is less than some maximum. To complicate matters further, the players are free to switch between teams at will. Furthermore, the soccer field does not have to be symmetrical: the field can be on an incline such that the ball naturally rolls towards one team's goal, and it can also include obstacles on one end that are not on the other. The pool of possible players also does not have a limit, ranging as high as 1,000, and the players are free to leave and join a match at any time. There are no other published models designed to handle this complex of a competition. The following presents a model to estimate individual ratings from team-play and account for both the imbalance in the play-style, and the dynamic nature of the teams.

With the models presented here, we are able to answer several questions, including, “Who are the best players on this server?”, “Which maps are the most difficult for the Axis side?”, and “Which server is easier to play on for the average player?” in addition to questions like “How likely are the Allies to win the current match on the this server?”

This paper will proceed as follows. Section 1.1 will give related work, section 2 describes the data analyzed, the models are presented in section 3, section 4 explains how the model parameters are estimated, section 5 shows the results of the estimation, and section 6 gives conclusions and directions for future work.

1.1 Related work

The basic model used in this paper is an extension of the commonly used model proposed by Bradley and Terry (1952) for paired-comparisons. In this model, two opponents have ability parameters λ_1 and λ_2, and the probability of the first opponent winning is $\lambda_1/(\lambda_1 + \lambda_2)$. Several reviews on uses of and extensions to Bradley-Terry models can be found in the literature. (Bradley and Terry, 1952; Davidson and Farquhar, 1976; David, 1988; Hunter, 2004) One common extension proposed by Agresti (1988) is to add a parameter to the model for home field advantage. The following uses a similar approach, but instead of a single parameter, there are two per map—one for the Axis side and one for the Allies side.

What makes this work unique is that it presents a new model for estimating individual ratings from team competitions. Few other models have been presented for this situation. One notable model is that of Huang et al. (2006), who combine the ratings of a team by adding the ratings of its players. This model fits well in their application which never had uneven amounts of individuals per group. It results in a linear increase in team rating when there are more players on one team than another. Experience has shown us, however, that having more players on one team does not result in a linear, but rather exponential increase in the team’s perceived ability to win. Another similarity between our model and that of Huang et al. (2006) is that we both present a method for weighting individuals by how much they are expected to contribute to their respective teams. Huang et al. (2006) suggest the weighting be predetermined using expert knowledge about the data, whereas we use a simple exposure model that uses the percent time a player plays on a given team for a given match as that player’s “weight” for that match.

2 Data

We obtained data by parsing the log files from three Enemy Territory servers yielding around 100 matches per server. For each match, the data lists which server the match was played on, which map was played, which side won—Axis or Allies—the length of the match, which players participated, and how long each player spent on both teams.

3 Chosen Models

The following shows how we used the given data to estimate player ratings. We first present models for the individual player ratings, and then the likelihood model for predicting whether a given team will win a match given the data above.

3.1 Fundamental Model

Letting $\theta_{i,m-s}$ represent player i's ability when playing on the map-side combination $m - s$, a simple model for the player's ability is:

$$\theta_{i,m-s} \equiv \theta_i \text{ and } \theta_i \sim N(\mu, \sigma_\theta^2). \tag{1}$$

σ_θ^2 is given a prior distribution and $\mu = 0$ without loss of generality. This model assumes that each player has a single ability parameter that is unaffected by which side they are playing on, and which map they are playing on.

3.2 Accounting for Map-Side Effects

Because maps in Enemy Territory and most other games of this nature are so varied, it is convenient to describe the details of a given match in terms of the combination of map and side that each player played on. Throughout this paper, we will refer to a team or a player's "map-side" combination, meaning the map the match took place on, e.g. the map "venice", and the side the player played on—Axis or Allies. The model given above is naive considering most maps were designed such that one side has a major advantage. Since this imbalance is uniform by design for all players, a more robust model that takes into account the side being played on is given as follows:

$$\theta_{i,m-s} \equiv \theta_i + \gamma_{m-s} \text{ with } \theta_i \sim N(0, \sigma_\theta^2) \text{ and } \gamma_i \sim N(0, \sigma_\gamma^2). \tag{2}$$

This is similar to fitting the "homefield advantage parameter" proposed by Agresti (1988), except that instead of one parameter per "field", here there are two—one for the Axis side, and one for the Allies. This is because it is not clear from the data which side has the advantage, and so we need to infer this by fitting a parameter for both sides. Although this does not model individual player-map-side interactions, it is assumed that the additive effect will be stronger than the individual interaction effects because most maps have been designed to be uniformly uneven. Since we are more interested in the predictive power in the ratings than the interactions themselves, we chose not to model them at this point.

One of the interesting effects of this model is that the estimates for $\gamma_{m,s}$ can be used to judge which maps are more balanced balance between Axis and Allies. Server administrators can use this information to choose which maps to place on their servers to improve gameplay, and map creators can use this information to better balance the gameplay in their maps.

3.3 Server Easiness

If players can be ranked based on matches they participated in on a given server, it would be beneficial to also be able to compare players participating on different servers, or players that played on several servers. In order to determine how a given server affects a player's rating, the following model adds a server bias into each player's rating, yielding:

$$\theta_{i,m-s,j} \equiv \theta_i + \gamma_{m-s} + \psi_j \text{ with } \theta_i \sim N(0, \sigma_\theta^2), \gamma_i \sim N(0, \sigma_\gamma^2) \text{ and } \psi_j \sim N(0, \sigma_\psi^2). \tag{3}$$

With this model, a player's rating depends on their base ability, the side and map they are playing on, and the server they are playing on. Besides being able to compare players across servers more effectively, estimating ψ also gives us a measure of how difficult each server is. This information can be used by newer players to choose easier servers, or more experienced players can use it to find more challenging gameplay. In addition, since the player ratings can now be fit globally over all servers, a player can compare his or herself to all of the players in the game, instead of only those who play on a certain server. The accuracy of the comparison, however, will be affected by how often players move between servers. If enough server "cross-over" occurs, then the ratings should be reasonable enough for comparison.

Another interesting aspect of ψ is since the amount added to each side is equal to $N_s\psi$ where N is the number of players on side s, ψ can be interpreted as the rating increase a given side s expects for each additional player on that side. Servers where having additional players will not make up for the skill of the players are more difficult than those where a few extra players alone can decide the winner.

For this paper, we ranked players both with and without using ψ in order to determine their standings compared to only those players on the respective servers, and their abilities overall.

3.4 Likelihood

The likelihood model we use is based on a modified version of the Bradley-Terry paired-comparison model (Bradley and Terry, 1952). In order to predict which of two teams is more likely to win, we choose a side s's probability of winning a match played on map m to be proportional to $\lambda_{s,m} = exp(\sum_{i=1}^{|P_s|} \theta_{i,m-s})$ where $|P_s|$ is the number of players on side s. Therefore the probability of side s_{Axis} defeating side s_{Allies} for the given map is $\lambda_{s_{Axis},m}/(\lambda_{s_{Axis},m} + \lambda_{s_{Allies},m})$. This results in the probability of a given side winning increasing exponentially with the size of that side compared to the other. This is appealing because unlike a game like chess where only one move can be made per side despite the number of actual players making the move decisions, in Enemy Territory and other real-time online team competitions, every player can act simultaneously. Having more players gives a team with more numbers the ability to more effectively defend or accomplish its objectives. It is rare for a team with two or less players than another to win unless that team is very good.

If G is the total number of matches, $w(g)$ gives the winning side for match g, $l(g)$ the losing side, and m the map played on, then this model gives rise to the following likelihood function:

$$P(w|\lambda) = \prod_{g=1}^{G} \lambda_{w(g),m}(\lambda_{w(g),m} + \lambda_{l(g),m})^{-1} \tag{4}$$

where

$$\lambda_{w(g),m} = exp(\sum_{i=1}^{|P_{w(g)}|} (\theta_{i,m-s})) \tag{5}$$

and

$$\lambda_{l(g),m} = exp(\sum_{i=1}^{|P_{l(g)}|} (\theta_{i,m-s})). \tag{6}$$

One problem with the above likelihood is that in public-style servers players can come, leave, and change teams almost at will. Therefore, a model that does not take into account how much time a player spends on each team is inaccurate. Since the match data we are using does indicate the amount of time the players spent per team, this information can be used to modify the λ's above:

$$\lambda_{w(g),m} = exp(\sum_{i=1}^{|P_{w(g)}|} (\tau_{i,w(g)}\theta_{i,m-s})) \quad (7)$$

$$\lambda_{l(g),m} = exp(\sum_{i=1}^{|P_{l(g)}|} (\tau_{i,l(g)}\theta_{i,m-s})) \quad (8)$$

where $\tau_{i,w(g)|l(g)}$ is the percent of the total match time player i spent on the winning team, $w(g)$, and the losing team $l(g)$. This yields a simple approximation to integrating the player abilities over the time of the match to yield the overall strengths λ, per team.

4 Analysis strategies

This section will discuss how the priors were chosen, how the model was fit, and how the convergence of the fit was verified.

4.1 Prior selection

The models presented here will have at most three priors. There is a prior on the standard deviation of the player ratings, σ_θ^2, a prior on the standard deviation of the map-side effects σ_γ^2, and a prior on the server effects σ_ψ^2. Instead of choosing non-informative priors for these standard deviations, we placed hyperprior distributions on them using inverse gamma distributions, with α and β chosen such that the distributions have means of 1.0 and variances at 1/3. This was done simply to keep most player ratings between -3 and 3, without loss of generality.

4.2 Software

Software to fit the models used in this paper was written in Python specifically for these analyses. It uses MCMC with Gibbs steps to sample from the standard deviations and Metropolis steps to sample from the player rating θs, the map-side effect γs, and the server easiness ψs. This leaves a potentially large set of tuning parameters, but following the suggestion of Graves et al. (2003), we chose separate constants for both player- and map-step sizes and divided them by the square root of the number of games a player had played in, or the number of times that map had been played. This resulted in acceptance rates of around 40% for the parameters fit using Metropolis steps.

4.3 Convergence Diagnostics

While it is difficult to guarantee an assessment of the mixing and convergence for MCMC, time series plots and acceptance rate analyses were used and the criteria in the diagnostics proposed by Raftery and Lewis (1996) were met.

5 Results

In this section we derive rankings for players based on match information on three different Enemy Territory servers. First, rankings are derived for each server separately, and then we add an additional model parameter to estimate the easiness of each server. This second model is then used to rank the players across all three servers simultaneously. In addition, the difficulty of the maps with respect to which side a player chooses are analyzed.

5.1 Separate Server Rankings

In this section we present the top ten rankings of the players per server applying the model without using the server effect parameter ψ. The model was fit once per server using matches from only that server. Each time, 20,000 iterations of MCMC were used with 10,000 iterations of burn-in. The results are shown in tables 1–3. The players are conservatively ranked by how they play two standard deviations below their posterior means. This has the desired effect of penalizing players who have not played as many games. It is not uncommon in these tables to see players who have worse winning percentages ranked higher than those with better ones. For instance, in table 1, BlackSheep has a higher posterior mean rating and is ranked higher than 0c3DiGiTaL despite having an 8-1 record vs. 0c3DiGiTaL's 9-0 record. This suggests that BlackSheep's wins were against more difficult odds—e.g. harder map-side combinations or against better players.

Since the players who play commonly on public servers rarely receive popular attention, the aliases used will not likely be recognizable to the general public. However, the goals of giving each player a rating and hence a stronger incentive to continue playing and improving his or her abilities in the game, in addition to better predicting the outcome of the games, have been met.. That said, the players shown are known to be highly skilled amongst those who play on the servers.

Table 1: Ratings and rankings for server 1. Shown are the player ranks, the posterior means of their base ratings, θ_i, the standard deviation of each player's θ_i, the nicknames they in-game, the number of games each played, the number of times they won, and the number of times they lost.

Rank	Post. Mean	SD	Name	Games	Wins	Losses
1	1.58	0.90	K4F*Boas7	39	26	13
2	1.19	0.87	|R!P|Burnz	27	21	6
3	0.74	0.74	|R!P|Crazyeskimo	52	30	22
4	1.15	1.07	BlackSheep	9	8	1
5	0.75	0.90	=MG=.:Guen:.	26	15	11
6	1.11	1.08	0c3DiGiTaL	9	9	0
7	1.08	1.07	Ignis	7	7	0
8	0.80	0.98	<ETA>Ronik	17	10	7
9	0.52	0.84	-|S*S|-dgifTy!	28	17	11
10	0.75	0.97	sliveR	12	9	3

Table 2: Ratings and rankings for server 2.

Rank	Post. Mean	SD	Name	Games	Wins	Losses
1	2.65	1.62	Sambuku	25	18	7
2	1.27	1.20	UruhA	32	22	10
3	1.43	1.32	-=DM=- nixx(cz)	18	13	5
4	1.51	1.40	K4F*SHERREMAN	11	9	2
5	1.05	1.25	<ETK>Soldier!Ryan	27	16	11
6	1.26	1.39	LaMigra	23	16	7
7	1.31	1.47	PaRaDiCe	14	11	3
8	1.05	1.36	Randy	29	18	11
9	1.29	1.56	SMG.Goku.	17	14	3
10	0.99	1.42	|R!P|Th@iN	13	8	5

Table 3: Ratings and rankings for server 3.

Rank	Post. Mean	SD	Name	Games	Wins	Losses
1	2.52	1.57	SeXyBloodLust	42	26	16
2	2.13	1.38	b!t	52	32	20
3	2.63	1.86	Pizza Toby	11	10	1
4	2.20	1.65	Sexy Hu$tle	14	11	3
5	1.65	1.47	Assassin	23	14	9
6	1.07	1.35	SEXY SQUIDDY	46	25	21
7	1.92	1.80	Smaug	18	13	5
8	1.24	1.48	JasonBourne	28	12	16
9	1.32	1.55	Lispnik	17	11	6
10	1.81	1.80	Sexy* Wikked Man	18	14	4

Table 4: Ratings and rankings for all three servers combined.

Rank	Post. Mean	SD	Name	Games	Wins	Losses
1	3.02	1.26	K4F*Boas7	66	43	23
2	3.60	1.78	Sambuku	25	18	7
3	2.90	1.45	SeXyBloodLust	42	26	16
4	3.10	1.56	|R!P|Burnz	27	21	6
5	1.93	1.11	b!t	74	41	33
6	3.27	1.86	Pizza Toby	11	10	1
7	3.36	1.94	BlackSheep	9	8	1
8	3.18	1.85	|R!P|HunTeR	15	12	3
9	1.76	1.39	UruhA	45	30	15
10	1.20	1.12	|R!P|Crazyeskimo	70	41	29

Table 5: Server easiness.

Rank	Post. Mean	SD	Server
1	3.00	0.90	1
2	2.00	0.82	3
3	1.42	0.73	2

5.2 Combining all 3 Servers

To determine which were the best players over all three servers, we took advantage of the fact that many players play on more than a single server in order to fit the server easiness parameter, ψ. Again, the model parameters were estimated using 20,000 iterations of MCMC with 10,000 iterations of burn-in. The results are shown in tables 4–5.

The server results are surprising at first because most players consider the settings on server 1 to make it a more difficult server. However, the fact that the matches on server 1 are generally smaller because it is the least popular of the servers may contribute to it being easier overall for those players who play on multiple servers. It is not as surprising to see server 2 above server 3 since server 2 is the most popular server and attracts many of the best players. It is interesting the number of top players that do not come from server 2's top ten list, considering it should be the harder server. This suggests those that are in the top performed exceptionally across all servers despite the easiness of the servers.

5.3 Map-Side Effects

One of the interesting effects of the model used is that rating parameters are also fit for each map-side combination. This information can be used to judge which maps are more even and which maps have the

Table 6: Map-side ratings for first the most uneven map and then the most even map on server 1. The row show the in-game name of the map, the side of the map described by the current row, the posterior mean of the map's rating or γ for that side, and the standard deviation of that map's rating.

Name	Side	Post. Mean	SD
oasis	Allies	0.6313	0.561
oasis	Axis	-0.5131	0.6188
fueldump	Allies	0.36	0.4959
fueldump	Axis	-0.06	0.5338

Table 7: Map-side ratings for first the most uneven map and then the most even map on server 2.

Name	Side	Post. Mean	SD
oasis	Allies	0.7088	0.5831
oasis	Axis	-0.6387	0.618
venice	Allies	0.1973	0.5086
venice	Axis	0.1501	0.5527

least balance between Axis and Allies. Tables 6–9 show the results of fitting the map-side parameters on each server for first the most uneven map on that server, and then the most even map. In general, the results are in line with the perception of the players. Both the *oasis* and *radar* maps are known to be a difficult for the Axis side to "defend", whereas *fueldump*, *venice*, and *goldrush* are known for their fairness and commonly listed as player favorites.

From these findings, it is not unreasonable to conclude that, for example, oasis is imbalanced in favor of the Allies and that venice is probably a fair map. Server administrators could decide to either not include maps like oasis in the list of maps on their server, or they can take measures to ensure that when oasis is played, the Axis team consists of more skilled players. Map makers can use this information to consider whether they should make changes to oasis that would make it more even given equally skilled teams. Games with statistically driven level design are more likely to attract players, as are servers that include more balanced maps in their lineups.

Table 8: Map-side ratings for first the most uneven map and then the most even map on server 3.

Name	Side	Post. Mean	SD
radar	Allies	0.4989	0.5335
radar	Axis	-0.288	0.549
goldrush	Allies	0.141	0.548
goldrush	Axis	-0.0473	0.5874

Table 9: Map-side ratings for first the most uneven map and then the most even map for all three servers combined. These come from the model fit using ψ.

Name	Side	Post. Mean	SD
oasis	Allies	0.2486	0.2051
oasis	Axis	-0.3684	0.252
venice	Allies	0.0326	0.1868
venice	Axis	-0.1328	0.1998

5.4 Measuring Performance

The performance of the models was measured by their accuracy in predicting the matches used to estimate the model parameters. This results in a positive bias for the results. An unbiased estimate of the models' accuracy would require computationally intensive methods such as leave-one-out cross-validation. These methods would be impossible in practice because the amount of time it takes to fit the model using MCMC is longer than the length of the average match. Therefore, a future method will be developed that approximates the given models more efficiently. That said, each set of parameters achieved near 100% accuracy in predicting the match data. This suggests that the ratings derived for the players are reasonably accurate over the given matches.

6 Conclusions and Future Work

We have presented new models for estimating individual ratings in team competitions. These ratings allow players to effectively track their ability to help the teams they play on win. Games and servers that provide well-developed methods for tracking their players' abilities are more likely to attract players and therefore be more profitable. The models presented are also able to account for both the dynamic nature of the teams in public, online team-competitions, and the imbalance commonly associated with the levels or maps designed for these competitions. In addition, we have estimated parameters that allow us to analyze the fairness of the maps in terms of the sides playing. This information can be use to create maps that are more fair and therefore more enjoyable for the players, or the information can be used by server administrators to choose different maps for their respective servers.

In addition, we have fit a parameter that allows us to estimate the "easiness" of each of a set of servers. This information can be used to allow players to select servers that better match their abilities. Players are more likely to continue playing a game if they can ensure matches will not be too easy or too difficult. The models developed in this paper can be directly applied to modern-day online team-competitions with results that will improve gameplay and attract players.

For future work, we note that the method used to estimate the model parameters in this paper, MCMC, is too computationally intensive to be used real-time in situations involving thousands of players and servers. Therefore, we will develop a more efficient method for determining player, map, and server ratings. This method needs to update the model parameters quickly after every competition on every server. One of the benefits of having a real-time approximation to these models is that the information can be used in real-time to improve the fairness of the matches.

For example, a system can be developed that uses the player and map information on a given server to automatically balance the teams currently playing using real-time rating information, without the need of a server administrator who has prior knowledge about player abilities. Such a system could ensure the teams playing are always reasonably fair by moving players from team to team when the likelihood of a team winning is very low. Servers and games that keep the gameplay well-balanced will be more successful.

In addition to automatically balancing the teams in a current match, a faster approximation to these models can provide real-time information about the easiness of the servers in a game, or even a real-time estimate of the probability a given player is better or worse than the players currently playing on each server. Players can use this information in-game to select servers that better match their abilities.

One potential problem with the models presented here is that they do not account for time-varying changes in player skill. Some players are known veterans whose abilities are not likely to change appreciably over time, however others are newer players whose abilities are quickly improving. These changes in ability can be modeled explicitly, but another solution is to design the aforementioned faster approximation as a recursive algorithm. A recursively updating algorithm will naturally account for changes in player performances over time.

References

Agresti, A., 1988. Categorical Data Analysis. Oxford University Press.

Bradley, R., Terry, M., 1952. The rank analysis of incomplete block designs I: The method of paired comparisions. Biometrika 39, 324–345.

Crandall, R. W., Sidakk, J. G., 2006. Video games: Serious business for america's economy. URL http://www.theesa.com/archives/files/2006%20WHITE%20PAPER%20FINAL.pdf

David, H. A., 1988. The method of paired comparisons. Oxford University Press.

Davidson, R. R., Farquhar, P. H., 1976. A bibliography on the method of paired comparisons. Biometrics 32, 241–252.

DFC Intelligence, 2006. Online game market.
URL `http://www.dfcint.com/game_report/Online_Game_toc.html`

Graves, T., Reese, C., Fitzgerald, M., 2003. Hierarchical models for permutations: Analysis of auto racing results. Journal of the American Statistical AssociationTo appear.

Huang, T.-K., Weng, R. C., Lin, C.-J., 2006. Generalized bradley-terry models and multi-class probability estimates. Journal of Machine Learning Research 7(Jan), 85–115.

Hunter, D. R., 2004. Mm algorithms for generalized Bradley-Terry models. The Annals in Statistics 32, 386–408.

Jarret, A., 2003. Igda online games white paper 2nd edition – march 2003.
URL `http://www.igda.org/online/IGDA_Online_Games_Whitepaper_2003.pdf`

Raftery, A., Lewis, S., 1996. Implementing MCMC. Chapman and Hall, pp. 115–130.

The Entertainment Software Association, 2006. Essential facts about the computer and video game industry.
URL `http://www.theesa.com/archives/files/Essential%20Facts%202006.pdf`