Rating Players in Games with Real-Valued Outcomes

(Extended Abstract)

Christopher Archibald, Neil Burch, and Michael Bowling
Department of Computing Science
University of Alberta
Edmonton, AB, Canada T6G 2E8
{archibal, nburch, bowling}@ualberta.ca

Matthew Rutherford
Department of Computer Science
University of Denver
Denver, CO 80208
mjr@cs.du.edu

ABSTRACT

Game-theoretic models typically associate outcomes with real valued utilities, and rational agents are expected to maximize their expected utility. Currently fielded agent rating systems, which aim to order a population of agents by strength, focus exclusively on games with discrete outcomes, e.g., win-loss in two-agent settings or an ordering in the multi-agent setting. These rating systems are not well-suited for domains where the absolute magnitude of utility rather than just the relative value is important. We introduce the problem of rating agents in games with real-valued outcomes and survey applicable existing techniques for rating agents in this setting. We then propose a novel rating system and an extension for all of these rating systems to games with more than two agents, showing experimentally the advantages of our proposed system.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Application and Expert Systems—*Games*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi Agent Systems*

General Terms

Experimentation, Economics, Measurement

Keywords

Player Ratings, Real-valued Games, Least Squares, Ridge Regression

1. INTRODUCTION

In games with real-valued outcomes the goal of players is not to win, but instead to *maximize* the utility gained during play. Existing rating schemes that are applicable have only been defined for the two-player case, and have only been applied in games (like American football or basketball) where the actual goal of the participants is simply to win, not actually to maximize their score. We examine the efficacy of those previous schemes in our setting and also introduce a new regularized least squares ratings system. We extend all of these rating systems to games with more than two players which have real-valued outcomes. All of these systems are experimentally evaluated in the domain of poker.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

2. THE PLAYER RATING PROBLEM

Our goal is to provide player ratings for a population of players who play a given zero-sum (not necessarily two-player) game. Let $S = \{1, 2, \dots m\}$ be the set of m seats, or positions, in the game. We denote the population of players by the set $N = \{a, b, \dots\}$, where $|N| = n$. For a given game G and a population of players N, we define a *game configuration* g to be the tuple (ϕ_g, u_g, k_g). The function $\phi_g : S \mapsto N$ denotes the mapping from game positions to players in the population N. The inverse mapping ϕ_g^{-1} : $N \mapsto S \cup \emptyset$ specifies the seat position, or not participating ($\emptyset$), that every player in N occupies in game configuration g. The function $u_g : S \cup \emptyset \mapsto \mathbb{R}$ describes the average utility, or average payoff, that the player in each seat of the game received as a result of participating in $k_g \in \mathbb{N}$ independent *game instances* or *game repetitions* of the specific arrangement of players composing this game configuration. The total amount of utility gained by a player a in a game configuration g will then be equal to $k_g u_g(\phi_g^{-1}(a))$. For ease of exposition we define $u_g(\emptyset) = 0$, that is, the net utility for any player not participating in that game configuration is 0. The *real-valued rating problem* is to take a dataset D of game configurations for a population N and to determine for each player $a \in N$ a single real-valued rating r_a which is a good fit to this data. We note that in this work we treat a player's rating as a static value and consider the k_g game instances of each game configuration to be independent and identically distributed samples drawn from a distribution with mean equal to the true expected utility u_g^* of each player in game configuration g.

3. RATINGS IN REAL-VALUED GAMES

We begin by introducing a simple linear model of how players' skill ratings yield the payoffs in the game. We will denote by $\delta : S \times S \mapsto [1, m] \subset \mathbb{N}$ a function which maps two seat positions in the game to an index which captures the relative position of these seats in the game. Our system will learn a vector of weights, denoted by β, which will contain a value for each possible output of the function δ. The β values, intuitively, are meant to capture the strength of the impact of players in each position on the net payoff of players in the other positions, including their own.

Given ratings for all participants in a game configuration g, the linear model will predict the expected net payoff p_a of player a in g to be $p_g(a) = \sum_{s \in S} \beta \left[\delta(\phi^{-1}(a), s)\right] r_{\phi(s)}$. Given the data, our goal is to determine values for the β weights and ratings r for the players such that the predicted outcome of each game configuration matches the observed outcome in the data. Each presented rating system uses a different method for determining "fit." Since the game is zero-sum, for any $j \in S$, $\sum_{s \in S} \beta \left[\delta(j, s)\right] = 0$. This reduces the number of β values we need to search over, since we

Table 1: Optimization Objectives

LS	$\sum_{g\in D}\sum_{a\in N} k_g \left(\left(\sum_{s\in S}\beta\left[\delta(\phi_g^{-1}(a),s)\right] r_{\phi_g(s)}\right) - u_g(\phi_g^{-1}(a))\right)^2$
LAV	$\sum_{g\in D}\sum_{a\in N} k_g \left|\left(\sum_{s\in S}\beta\left[\delta(\phi_g^{-1}(a),s)\right] r_{\phi_g(s)}\right) - u_g(\phi_g^{-1}(a))\right|$
RLS	$\sum_{g\in D}\sum_{a\in N} k_g \left(\left(\sum_{s\in S}\beta\left[\delta(\phi_g^{-1}(a),s)\right] r_{\phi_g(s)}\right) - u_g(\phi_g^{-1}(a))\right)^2 + \sum_{b\in N}(\gamma r_b)^2$

actually only have to determine the value of $|S|-1$ of the β values, and the last value will be equal to zero minus the sum of the previous values. We also choose to fix the β scale so that the skill ratings of the players are in the same units as the payoff of the game. We do this by fixing $\beta\left[\delta(s,s)\right]$ (the weight for the seat of the player whose payoff we are predicting) to be 1, for all $s \in S$. With β constrained in this manner, there is a natural way to interpret the meaning of the player ratings. Specifically, a player's rating corresponds to the expected amount of utility they would be predicted to gain if all of their opponents in a game configuration had a rating of zero.

3.1 Rating Approaches

Each approach we evaluate differs in how it measures the "fit" of a set of predictions to the actual game configuration outcomes in the data. The use of squared error (LS) was introduced for 2-player games in 1977 [5] and 1980 [4], where it was used to rank teams in sports like American football and basketball and focused on predicting the actual scores of games. The use of absolute error (LAV) was introduced and evaluated, again in the two-team game case, in 1997 [2]. This previous work always involved only two teams and was only ever applied to games where the main goal of teams is (arguably) to win, not maximise score. In the settings for which these previous methods were introduced, teams being rated had participated in similar number of games. In general, there is no reason why players will necessarily have participated in the same number of game instances, and when data is sparse, the ratings will "overfit." Obviously, the more game instances that a player participates in, the more evidence we have of their actual skill level. This is the idea behind our novel regularized least square rating system (RLS). It attaches a penalty to large ratings, so ratings will be close to zero unless there is enough evidence in the data to show that they should be something else.

The optimization objective for each method is given in Table 1. In each case the ratings r and β values are optimized, in order to minimize the given objective function, using Alternate Convex Search (ACS) [3]. This algorithm is defined for the optimisation of biconvex functions. In our setting, we initialize each β entry to be $\frac{-1}{|S|-1}$. We then find the optimal ratings r, given that β. Then we find the optimal β, given those ratings r. This process is repeated until convergence. Exact methods (normal equations or LP solvers) are employed to optimise r and β in each step. Of course, in the two-player case ($|S| = 2$), there are no β values to optimise, so the globally optimal ratings can be computed. In the general case ACS is guaranteed to find only a local, not a global, optimum.

4. EXPERIMENTS

We use the data from the 2012 Annual Computer Poker Competition's (ACPC) two-player limit (2P-L), two-player no limit (2P-NL), and three-player limit (3P-L) events, which is available for download on the ACPC website [1]. For purposes of comparison, we also include the results for a zero-rating scheme in each experiment. This rating system is completely data-agnostic and rates all players the same, predicting that every game instance will end with each player receiving a payoff of zero. We divide the data into training and test sets in the following manner. We begin with an empty set of training game configurations and repeatedly select a random game configuration from the data and add it to the training set. After each new game configuration is added, we check to determine whether the current training set of game configurations connects the set of players, i.e. does a sequence of common opponents connect any two players in the population so that the ratings obtained are meaningful. Once the growing set of game configurations is sufficient to connect all the players, the process stops, and this becomes the training set. The test set is simply the remainder of the game configurations. The scores reported are averaged over 100 training and test sets created randomly in this manner. This evaluation method creates, in some sense, the minimum number of game instances necessary to rate the players in the population.

We use both the *Mean squared error (MSE)* and the *Mean absolute error (MAE)* to evaluating the quality of a particular set of ratings on the test set. Each method weights the error of the ratings' outcome prediction in each game configuration g by k_g, the number of game instances that compose this game configuration.

The regression coefficient for the RLS ratings was optimized on the corresponding 2011 ACPC data. We see that by both metrics, the RLS ratings achieved the lowest average error on each dataset.

Table 2: Rating Accuracy on 2012 ACPC Data

MSE						
Data	Zero	Mean	Median	LS	LAV	RLS
2P-L	0.0666	0.0394	0.0428	0.0131	0.0138	**0.0107**
2P-NL	36466	62340	59739	81543	81507	**34886**
3P-L	0.0303	0.0070	0.0090	**0.0023**	0.0028	**0.0023**
MAE						
Data	Zero	Mean	Median	LS	LAV	RLS
2P-L	0.2058	0.1516	0.1506	0.0868	0.0887	**0.0790**
2P-NL	108.68	151.15	133.01	171.31	157.93	**98.554**
3P-L	0.1473	0.0662	0.0765	0.0397	0.0442	**0.0371**

5. REFERENCES

[1] Annual Computer Poker Competition. Competition results. http://www.computerpokercompetition.org/, 2012.

[2] Jr. Bassett, Gilbert W. Robust sports ratings based on least absolute errors. *The American Statistician*, 51(2):pp. 99–105, 1997.

[3] J. Gorski, F. Pfeuffer, and K. Klamroth. Biconvex sets and optimization with biconvex functions - a survey and extensions. *Mathematical Models of Operations Research*, 66:373–407, 2007.

[4] David Harville. Predictions for national football league games via linear-model methodology. *Journal of the American Statistical Association*, 75(371):pp. 516–524, 1980.

[5] Raymond T. Stefani. Football and basketball predictions using least squares. *IEEE Transactions on Systems, Man and Cybernetics*, February 1977.