Documentation of the GGP Rating System

Martin Günther

January 21, 2008

Abstract

This document describes the current implementation of the GGP rating system.

1 Data structures

First, some words on the used data structures (ref. table 1). The data structures used for player, game and match have the intuitive meaning, but there is a new data structure called "match set". A match set simply is a collection of matches, all played in the same year, round and day of a competition, and all on the same game. All past competitions have featured such match sets, and explicit handling of match sets makes it possible to process all matches of such a match set in one batch.

Another unusual entry is the "coefficients" property of a game, which will be explained in section 2.1. For now, it suffices to think of it as the information that the linear regression rating stores about a game.

2 Linear regression rating

The central idea of linear regression rating is summarized in an email by Jim Clune:

> Here's some notation:
>
> | | |
> |---|---|
> | $g(m)$: | game associated with match m |
> | $s(r, m)$: | score of role r in match m |
> | $p(r, m)$: | player assigned to role r in match m |
> | $q(p)$: | rating of player p |
>
> It seems to me that what we want to be able to express $E[s(r, m)]$ (the expected value of the score of role r in match m) in terms of $q(p(r', m))$ for each role r' in the game. One way to achieve this might be to assume $E[s(r, m)]$ is a linear function of these variables and perfrom least-squares linear regression. For a game with roles $= \{r_1, r_2, r_3\}$, this yields relationships such as:

$$E[s(r_1, m)] = c_0 + c_1 * q(p(r_1, m)) + c_2 * q(p(r_2, m)) + c_3 * q(p(r_3, m))$$

class	property	description
player	rating	a single number reflecting the player's current rating (e.g. 1000.0)
game	name	the game name (e.g. chess)
	roles	the roles of the game (e.g. black, white)
	coefficients	linear-regression-specific game info
match	match ID	the ID of the match (e.g. Match.3390056123)
	players	the participants of the match *(→ player)*
	scores	the scores $(0 \dots 100)$ achieved by each player
	match set	the match set that this match is part of *(→ match set)*
match set	match ID	the ID of this match set (e.g. MatchSet.385572910)
	year, round, day, match set number	self-explanatory (e.g. 2007, 3, 2, 10)
	game	the played game (by definition, this is identical for all played matches of a match set) *(→ game)*

Table 1: Data structures

where c_0, c_1, c_2, c_3 are game-specific constants computed by the linear regression.

We could then calculate the expected outcome for each player in a given match and compare it with their actual outcome. If expected and actual outcomes are the same, the rating remains unchanged. Otherwise, the rating is adjusted up or down proportional to the difference between the actual outcome and the expected outcome.

- Jim

The following sections will show how to update the game-specific coefficients c_0, c_1, c_2, c_3 (section 2.1), how to calculate the expected scores of a player (section 2.2) and how to update the player ratings (section 2.3).

2.1 Updating the game information

The linear regression rating algorithm keeps so-called "game information" about each played game. This game information is just a matrix of linear regression coefficients and reflects the relationship between the role a player plays, the ratings of all players in the game and the expected score of the player in a certain game.

As stated above, linear regression rating assumes that the expected score of a player playing role $target\_role$ can be approximated by a linear function of the form:

$$E[s(r_1, m)] = c_0 + c_1 * q(p(r_1, m)) + c_2 * q(p(r_2, m)) + c_3 * q(p(r_3, m)) \quad (1)$$

The coefficients $c_0, \ldots, c_3$ are updated immediately after each played match set for the whole batch of matches. (The coefficients are different for each role, so the game information for a 3-player game such as this would consist of $3 * 4 = 12$ coefficients.)

If, in a given match set, players with a high rating scored worse (on average) than players with a high rating, the coefficient associated with the target role (c_1 in the example above) can become negative. If this is the case, the linear regression is repeated without the target role, and the target role's coefficient is set to zero. There are no constraints on the other coefficients.

If there are already known coefficients from former matches, the weighted average of the old and new coefficients is stored.

2.2 Calculating the expected scores

In the next step, the expected score of each player is calculated using formula 1. If the calculated value is less than 0 or greater than 100, the actual expected value is set to 0 or 100, respectively. After that, the total expected scores of each player are calculated by summing up the player's expected scores in all matches that the player played in that match set, so that the total expected score can exceed 100.

2.3 Updating the player ratings

In a third step, the player rankings are updated according to the following formula:

$$q'(p) \leftarrow q(p) + l * (a(p) - e(p)) \tag{2}$$

where $q(p)$ and $q'(p)$ are the current and updated rating of player p, l is the learning rate (cf. section 2.4), $a(p)$ is the actual and $e(p)$ the expected score. Update also takes place after each completed match set, so the player ratings are adjusted as quick as possible in order to provide more accurate information to the next game information updating step.

On a side remark, the player's ratings are initialized with a constant value (1000, at the moment).

2.4 Constant vs. dynamic learning rate

The choice of the learning rate has a big impact on the actual results of the algorithm. If the learning rate is too large, the player ratings become erratic and don't converge. If the learning rate is too small, convergation takes too long. One solution is to use a dynamic learning rate: starting off with a high learning rate and reducing it over time.

3 Results

The results for one run of both algorithms are shown in figures 1 and 2. Figure 3 shows the actual scores for comparison.

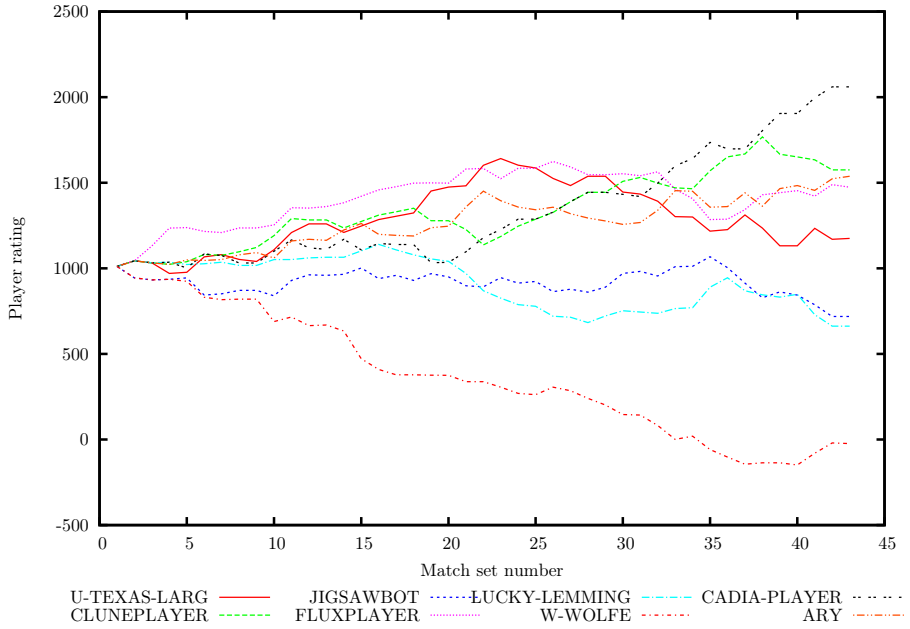

Figure 1: Constant linear regression ratings (Competition 2007 Preliminaries, constant learning rate = 1.0)

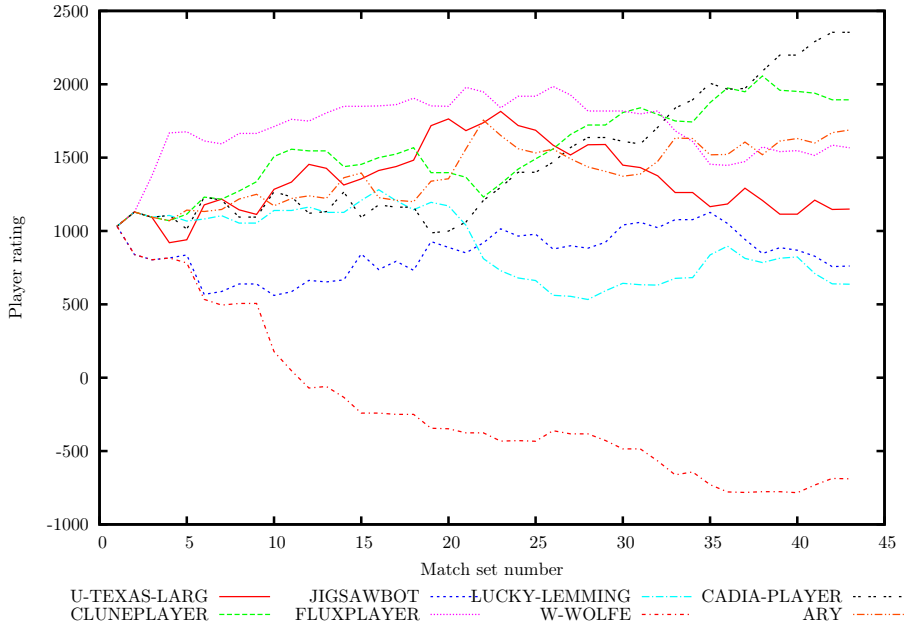

Figure 2: Dynamic linear regression ratings (Competition 2007 Preliminaries, dynamic learning rate = 6.0...1.6)

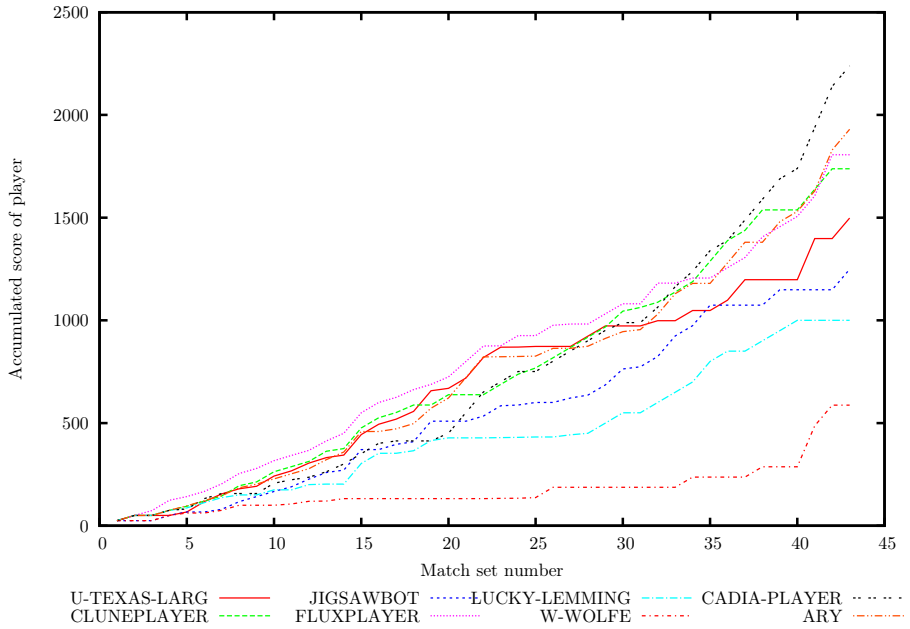

Figure 3: Direct scores (Competition 2007 Preliminaries; using round weights 0.25, 0.5, 0.5 and 1.0)