

# Elo-R Draft

December 5, 2015

Assume a collection of players labeled  $1, \dots, N$ . Player  $i$  has unknown skill  $s_i$ ; our objective is to estimate it by a *rating*  $r_i$ . The kernel of our Bayesian belief on  $s_i$  will be a product of logistic pdfs, which can be thought of as a prior times a set of independent measurements centered on  $s_i$  with different variances.

The players take part in a ranked event such as a video game challenge or programming contest. We model this as follows: each player  $i$  performs at level  $p_i$ , drawn independently from a logistic distribution centered at  $s_i$ . The ranking we observe is a total ordering on the players based on their performances:  $i$  outranks  $j$ , written  $i \succ j$ , iff  $p_i > p_j$ . According to this model, ties occur with probability zero; we will treat actual ties as half a win and half a loss.

Fix  $i$ . Let  $e$  be the evidence consisting of, for each  $j$ , whether or not  $i$  beat  $j$ . That is, we ignore the relative ordering of player pairs which don't include  $i$ . Our goal is to approximate the posterior distribution of  $s_i$  given  $e$ :

$$f(s_i | e) \propto f(s_i) \Pr(e | s_i) = f(s_i) \int \Pr(e | p_i) f(p_i | s_i) dp_i \quad (1)$$

Since the integral is hard to evaluate, we will treat  $\Pr(e | p_i)$  as a delta-function that spikes at the *maximum a posteriori* (MAP, a.k.a. posterior “mode”) estimate of  $p_i$ . This is justified as  $N \rightarrow \infty$ , because the evidence  $e$  would overwhelmingly concentrate  $p_i$  into a narrow range. Having fixed  $p_i$ , Equation (1) simplifies to

$$f(s_i | e) \propto f(s_i) f(p_i | s_i) \quad (2)$$

Our update algorithm thus divides into three phases: first, it must determine the player's performance  $p_i$  in the contest. Then, it will use this value to update the belief distribution on  $s_i$ . Finally, the belief distribution is summarized by finding the point  $r_i$  that achieves its maximum.

## 0.1 Performance estimation

To compute the MAP of  $p_i$ , we must maximize

$$f(p_i | e) \propto f(p_i) \Pr(e | p_i) \quad (3)$$

$p_i$  can be written as the sum of two logistic random variables. If we replace these by normal random variables with the same mean and variance, then their sum is also normal, and we approximate

it by a logistic with the same mean and variance. Thus if  $\alpha = \pi/\sqrt{3}$  and  $\tau = \sigma_i^2 + \delta^2$ ,

$$f(s_i) = \frac{2e^{2(s_i-r_i)/\sigma_i}}{\sigma_i (1 + e^{2(s_i-r_i)/\sigma_i})^2} \quad (4)$$

$$f(p_i | s_i) = \frac{2e^{2(p_i-s_i)/\delta}}{\delta (1 + e^{2(p_i-s_i)/\delta})^2} \quad (5)$$

$$f(p_i) \approx \frac{2e^{2(p_i-r_i)/\tau}}{\tau (1 + e^{2(p_i-r_i)/\tau})^2} \quad (6)$$

Let  $W_i = N - \text{Rank}_i$  denote the number of players  $i$  won against. Then

$$\Pr(e | p_i) = \prod_{j \succ i} \Pr(p_j > p_i) \prod_{j \prec i} \Pr(p_j < p_i) \quad (7)$$

$$\approx \prod_{j \succ i} \frac{1}{1 + e^{2(p_i-r_j)/\tau}} \prod_{j \prec i} \frac{e^{2(p_i-r_j)/\tau}}{1 + e^{2(p_i-r_j)/\tau}} \quad (8)$$

$$\propto \frac{e^{2W_i p_i / \tau}}{\prod_{j \neq i} 1 + e^{2(p_i-r_j)/\tau}} \quad (9)$$

$$C_1 + \ln f(p_i | e) = C_2 + \ln f(p_i) + \ln \Pr(e | p_i) \quad (10)$$

$$\approx \ln \frac{2}{\tau} + (p_i - r_i) \frac{2}{\tau} - 2 \ln \left( 1 + e^{2(p_i-r_i)/\tau} \right) + W_i p_i \frac{2}{\tau} - \sum_{j \neq i} \ln \left( 1 + e^{2(p_i-r_j)/\tau} \right) \quad (11)$$

Differentiate w.r.t.  $p_i$  and set to zero:

$$0 = \frac{2}{\tau} \left( 1 - \frac{2e^{2(p_i-r_i)/\tau}}{1 + e^{2(p_i-r_i)/\tau}} + W_i - \sum_{j \neq i} \frac{e^{2(p_i-r_j)/\tau}}{1 + e^{2(p_i-r_j)/\tau}} \right) \quad (12)$$

$$\text{Rank}_i = \frac{2}{1 + e^{2(p_i-r_i)/\tau}} + \sum_{j \neq i} \frac{1}{1 + e^{2(p_i-r_j)/\tau}} \quad (13)$$

Use binary search to solve for  $p_i$ . This is the *performance* of player  $i$  in the match.

## 0.2 Proportional rating update

Our approximate formula for the posterior  $f(s_i | e)$  takes the prior  $f(s_i)$  and multiplies it by a new logistic pdf  $f(p_i | s_i)$ . The general form for our posterior will be proportional to a product of normal and logistic pdfs. Since a product of normals is proportional to another normal pdfs, wlog we assume there is only one normal in the product:

$$e^{-(s_i-\mu_0)^2/\tau_0^2} \prod_k \frac{e^{2(s_i-\mu_k)/\tau_k}}{(1 + e^{2(s_i-\mu_k)/\tau_k})^2} \quad (14)$$

Differentiate its logarithm w.r.t.  $s_i$  and set to zero:

$$0 = \frac{2(\mu_0 - s_i)}{\tau_0^2} + \sum_k \frac{2}{\tau_k} \left( 1 - \frac{2e^{2(s_i - \mu_k)/\tau_k}}{1 + e^{2(s_i - \mu_k)/\tau_k}} \right) \quad (15)$$

$$0 = \frac{\mu_0 - s_i}{\tau_0^2} + \sum_k \frac{1}{\tau_k} \tanh \frac{\mu_k - s_i}{\tau_k} \quad (16)$$

Solve for  $s_i$  with binary search, and use its value as  $r_{i,new}$ .  $1/\sigma_i^2 = \sum_k 1/\tau_k^2$ .

Define the *information* contained in this distribution by  $I = \sum_k 1/\tau_k^2$ . The sum occurs by analogy to Gaussians: if we were to replace the logistic pdfs by Gaussian measurements with the same mean and variance, then their product would be a Gaussian with variance  $(\sum_k 1/\tau_k^2)^{-1}$ .

However, now we have to add noise to account for changing skills. Let  $\sigma^*$  denote a limit. Solve the fixpoint equation:

$$\frac{1}{(\sigma^*)^2} = \frac{1}{(\sigma^*)^2 + \nu^2} + \frac{1}{\delta^2} \quad (17)$$

$$(\sigma^*)^2 = \quad (18)$$

We say  $m$  is a weighted average of  $\{x_i\}$  with weights  $\{w_i\}$  if  $\sum_i w_i(m - x_i) = 0$ . In this sense,  $s_i$  is a weighted average of  $\mu_k$  with weights  $\tanh(\mu_k - s_i)/(\mu_k - s_i)$ .

### 0.3 Proportional Rating Update

If we approximate these by normal pdfs, their product is again normal, and we assign the corresponding mean and variance to our posterior belief over  $s_i$ :

$$r_{i,new} = \frac{\sigma_1^2 p_i + \sigma_2^2 r_i}{\sigma_3^2} \quad \sigma_{1,new} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_3^2} \quad (19)$$

Between contests, we model changes in a player's true skill by adding to  $s_i$  a Gaussian noise with mean 0 and variance proportional to  $\sigma_1^4/\sigma_3^2$ . Again approximating the product of pdfs by analogy to Gaussians, the noise conveniently resets our uncertainty  $\sigma_1$  to its original value.

### 0.4 Alternative rating update based on the MAP of $s_i$

$$C + \ln f(s_i | e) \approx \ln f(s_i) + \ln f(p_i | s_i) \quad (20)$$

$$= \frac{s_i - r_i}{\sigma_1} - \ln \sigma_1 - 2 \ln \left( 1 + e^{(s_i - r_i)/\sigma_1} \right) + \frac{s_i - p_i}{\sigma_2} - \ln \sigma_2 - 2 \ln \left( 1 + e^{(s_i - p_i)/\sigma_2} \right) \quad (21)$$

Differentiate w.r.t.  $s_i$  and set to zero:

$$0 = \frac{1}{\sigma_1} \left( 1 - \frac{2e^{(s_i - r_i)/\sigma_1}}{1 + e^{(s_i - r_i)/\sigma_1}} \right) + \frac{1}{\sigma_2} \left( 1 - \frac{2e^{(s_i - p_i)/\sigma_2}}{1 + e^{(s_i - p_i)/\sigma_2}} \right) \quad (22)$$

$$= \frac{1}{\sigma_1} \tanh \frac{r_i - s_i}{2\sigma_1} + \frac{1}{\sigma_2} \tanh \frac{p_i - s_i}{2\sigma_2} \quad (23)$$

Solve for  $s_i$  with binary search, and use its value as  $r_{i,new}$ . Note that if  $\sigma_1 < \sigma_2$ , then

$$|r_{i,new} - r_i| < \sigma_1 \ln \frac{\sigma_2 + \sigma_1}{\sigma_2 - \sigma_1} \quad (24)$$

In other words, this method enforces an upper bound on the rating change per event.

The first method can be thought of as setting  $r_i$  to the player's average historical performance, with exponentially decaying weights to place the emphasis on recent events. In contrast, the second method does something similar when the performances are consistent, but puts much less weight on distant outliers. Maybe this method can be made more accurate by remembering  $p_i$  values from the last 10 or so matches and computing the MAP on all 10 matches simultaneously with exponentially decaying weights, using the rating from 10 matches ago as the prior mean. The behavior induced by this modification can be likened to TopCoder's volatility measure: one very strong performance won't change the rating much, but the second or third consecutive strong performance will have a larger effect. Unlike on TopCoder, a very weak performance following a very strong performance will *not* have a large effect.