Elo-R Draft

November 20, 2015

Assume a collection of players labeled $1, \ldots, N$, each with unknown skill s_i drawn from a prior distribution logistic(r_i, σ_1^2). Thus, our prior knowledge about a player is parametrized by their *rating* r_i. They 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 logistic(s_i, σ_2^2). The ranking we observe is a total ordering on the players based on their performances.

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:

$$f(s_i \mid e) \propto f(s_i) Pr(e \mid s_i) = f(s_i) \int Pr(e \mid p_i) f(p_i \mid s_i) dp_i \tag{1}$$

Since the integral is hard to evaluate, we will treat $Pr(e \mid p_i)$ 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 $N \to \infty$, because the evidence e would overwhelmingly concentrate p_i into a narrow range. Having fixed p_i, Equation (1) simplifies to

$$f(s_i \mid e) \propto f(s_i) f(p_i \mid s_i) \tag{2}$$

0.1 Performance estimation

To compute the MAP of p_i, we must maximize

$$f(p_i \mid e) \propto f(p_i) Pr(e \mid p_i) \tag{3}$$

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 $\sigma_3^2 = \sigma_1^2 + \sigma_2^2$,

$$f(s_i) = \frac{e^{(s_i - r_i)/\sigma_1}}{\sigma_1 \left(1 + e^{(s_i - r_i)/\sigma_1}\right)^2} \tag{4}$$

$$f(p_i \mid s_i) = \frac{e^{(p_i - s_i)/\sigma_2}}{\sigma_2 \left(1 + e^{(p_i - s_i)/\sigma_2}\right)^2} \tag{5}$$

$$f(p_i) \approx \frac{e^{(p_i - r_i)/\sigma_3}}{\sigma_3 \left(1 + e^{(p_i - r_i)/\sigma_3}\right)^2} \tag{6}$$

Let $W_i = N - Rank_i$ denote the number of players i won against. Then

$$\begin{aligned} Pr(e \mid p_i) &= \prod_{j \succ i} Pr(p_j > p_i) \prod_{j \prec i} Pr(p_j < p_i) && (7) \\ &\approx \prod_{j \succ i} \frac{1}{1 + e^{(p_i - r_j)/\sigma_3}} \prod_{j \prec i} \frac{e^{(p_i - r_j)/\sigma_3}}{1 + e^{(p_i - r_j)/\sigma_3}} && (8) \\ &\propto \frac{e^{W_i p_i/\sigma_3}}{\prod_{j \neq i} 1 + e^{(p_i - r_j)/\sigma_3}} && (9) \end{aligned}$$

$$\begin{aligned} C_1 + \ln f(p_i \mid e) &= C_2 + \ln f(p_i) + \ln Pr(e \mid p_i) && (10) \\ &\approx \frac{p_i - r_i}{\sigma_3} - \ln \sigma_3 - 2 \ln \left(1 + e^{(p_i - r_i)/\sigma_3}\right) + \frac{W_i p_i}{\sigma_3} - \sum_{j \neq i} \ln \left(1 + e^{(p_i - r_j)/\sigma_3}\right) && (11) \end{aligned}$$

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

$$\begin{aligned} 0 &= \frac{1}{\sigma_3} \left(1 - \frac{2e^{(p_i - r_i)/\sigma_3}}{1 + e^{(p_i - r_i)/\sigma_3}} + W_i - \sum_{j \neq i} \frac{e^{(p_i - r_j)/\sigma_3}}{1 + e^{(p_i - r_j)/\sigma_3}}\right) && (12) \\ Rank_i &= \frac{2}{1 + e^{(p_i - r_i)/\sigma_3}} + \sum_{j \neq i} \frac{1}{1 + e^{(p_i - r_j)/\sigma_3}} && (13) \end{aligned}$$

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 $f(s_i \mid e)$ is a product of two logistic pdfs. 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:

$$r_{i,new} = \frac{\sigma_1^2 p_i + \sigma_2^2 r_i}{\sigma_3^2} \qquad \sigma_{1,new} = \frac{\sigma_1^2 \sigma_2^2}{\sigma_3^2} \tag{14}$$

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 σ_1^4/σ_3^2. Again approximating the product of pdfs by analogy to Gaussians, the noise conveniently resets our uncertainty σ_1 to its original value.

0.3 Alternative rating update based on the MAP of s_i

$$\begin{aligned} C + \ln f(s_i \mid e) &\approx \ln f(s_i) + \ln f(p_i \mid s_i) && (15) \\ &= \frac{s_i - r_i}{\sigma_1} - \ln \sigma_1 - 2 \ln \left(1 + e^{(s_i - r_i)/\sigma_1}\right) + \frac{s_i - p_i}{\sigma_2} - \ln \sigma_2 - 2 \ln \left(1 + e^{(s_i - p_i)/\sigma_2}\right) && (16) \end{aligned}$$

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

$$0 = \frac{1}{\sigma_1}\left(1 - \frac{2e^{(s_i-r_i)/\sigma_1}}{1+e^{(s_i-r_i)/\sigma_1}}\right) + \frac{1}{\sigma_2}\left(1 - \frac{2e^{(s_i-p_i)/\sigma_2}}{1+e^{(s_i-p_i)/\sigma_2}}\right) \tag{17}$$

$$= \frac{1}{\sigma_1}\tanh\frac{r_i-s_i}{2\sigma_1} + \frac{1}{\sigma_2}\tanh\frac{p_i-s_i}{2\sigma_2} \tag{18}$$

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

$$|r_{i,new} - r_i| < \sigma_1 \ln\frac{\sigma_2+\sigma_1}{\sigma_2-\sigma_1} \tag{19}$$

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 perforance will *not* have a large effect.