TrueSkill - Updating player skills in tennis with Expectation Propagation inference algorithm

Daniel Korzekwa (daniel.korzekwa@gmail.com)

March 21, 2013

What is the problem to solve?

Let's define a tennis player $p = \{player\_id\}$ and tennis match outcome $o = \{p_i, p_j, time, winner\}$. Given tennis players $\{p_1, ..., p_n\}$ and historical outcomes of tennis matches $\{o_1, ..., o_n\}$, predict the probability $p(o|p_i, p_j, t)$ of winning a tennis match by player p_i against player p_j at the time t.

Trueskill rating model

TrueSkill [1] is a Bayesian rating system developed by Ralf Herbrich, Tom Minka and Thore Graepel at Microsoft Reseach Centre in Cambridge, UK. Although, it is mostly used for ranking and matching players on Xbox Online Games, it is a general rating model that could be applied to any game, including Chess, Tennis or Football.

It models every player with a single skill variable $s \sim \mathcal{N}(x|m, v)$, which indicates how good player is on tennis. The expected skill value m is accompanied by a level of uncertainty v, which tells us, how confident we are about the player's skill estimation. Usually, the skill uncertainty decreases after observing the result of a game and it increases over the time, when player is not playing any games. Regarding to the expect skill value m, it moves up for a winner of a game and it shifts in an opposite direction for a loser.

What if we knew the true skill (variance $v = 0$) of a tennis player? Would we know for sure, how is he going to perform in a particular game? Probably not. It's because the player performance in a specific game depends on a number of factors, including player skill, player consistency, weather conditions and many other things. For that reason, we introduce performance variable $p \sim \mathcal{N}(x|m_s, v)$, with a variance v indicating the amount of uncertainty about player performance given his expected skill value m_s.

Let's introduce random variable $d \sim \mathbb{I}(p_i > p_j)$ that represents the difference between performance values for players p_i and p_j. Now, we can predict the outcome of a tennis match $o \sim \mathbb{I}(d > 0)$, which is the probability that player p_i will perform better in a game than player p_j. It is defined as $p(o) = 1 - \Phi_d(0)$, where $\Phi_d(0)$ is the value of a cumulative distribution function of a difference random variable d.

TrueSkill rating model is nothing else than a Bayesian Network, illustrated in a figure 1, composed of the random variables for skill s, performance p, performance difference d and match outcome o.

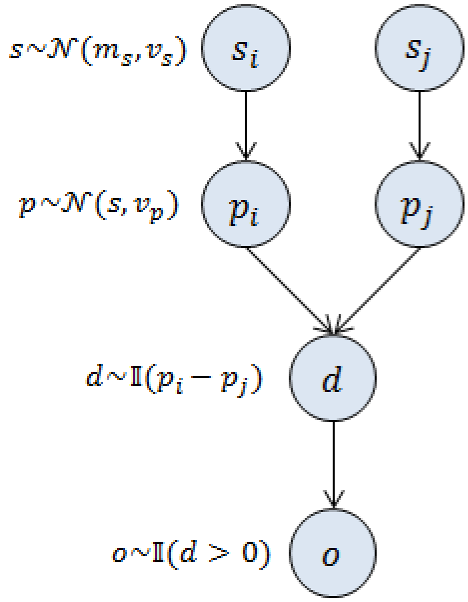

Figure 1: Bayesian Network for a True Skill rating model in tennis

Three queries of interest in the Tennis Bayesian Network include, predicting the outcome of a tennis match and computing marginal distributions for skill variables of both tennis players given observed outcome of a tennis game.

$$p(o) = 1 - \Phi_d(0) \tag{1}$$

$$p(s_i|o) = \int p(s_i, s_j|o)ds_j \tag{2}$$

$$p(s_j|o) = \int p(s_i, s_j|o)ds_i \tag{3}$$

Tennis example

Consider two tennis players p_1 and p_2 playing a tennis game and assume that we are provided with the probability distributions of skill and performance variables for both players at the beginning of the game.

$$\begin{aligned}
p(s_1) &= \mathcal{N}(x|m = 4, v = 81) \\
p(p_1|s_1) &= \mathcal{N}(x|m_s, v = 17.361) \\
p(s_2) &= \mathcal{N}(x|m = 41, v = 25) \\
p(p_2|s_2) &= \mathcal{N}(x|m_s, v = 17.361)
\end{aligned}$$

We ask for the probability of winning the game by player p_1 and we would like to know the skills for both players given player p_1 is a winner.

First, compute marginals of performance variables for both players.

$$p(p_1) = \int p(s_1)p(p_1|s_1)ds_1 = \mathcal{N}(m = 4, v = 98.368)$$
$$p(p_2) = \int p(s_2)p(p_2|s_2)ds_2 = \mathcal{N}(m = 41, v = 42.368)$$

Next, compute marginal for performance difference variable.

$$p(d) = \int p(p_1)p(p_2)\mathbb{I}(d = p_i > p_j)dp_1dp_2 = \mathcal{N}(x|m = -37.0, v = 140.736)$$

Now, compute the probability of winning the game by player p_1 at the beginning of a game.

$$p(o) = 1 - \Phi_d(0) = 0.0009$$

And finally, infer the skills for both players after the game and calculate the probability of winning the game given new skills.

$$p(s_1|o) = \int p(s_1, s_2|o)ds_2 = \mathcal{N}(x|m = 27.174, v = 37.501)$$
$$p(s_2|o) = \int p(s_1, s_2|o)ds_1 = \mathcal{N}(x|m = 33.846, v = 20.861)$$
$$p(o_{next}) = 1 - \Phi_{d\_next}(0) = 0.244$$

Figure 2 shows the skills for both players, before and after the game. The expected skill value increases for the winner of the game and it lowers for the loser. The value of variance around skills of both players goes down in a result of revealed information about the outcome of a tennis game.

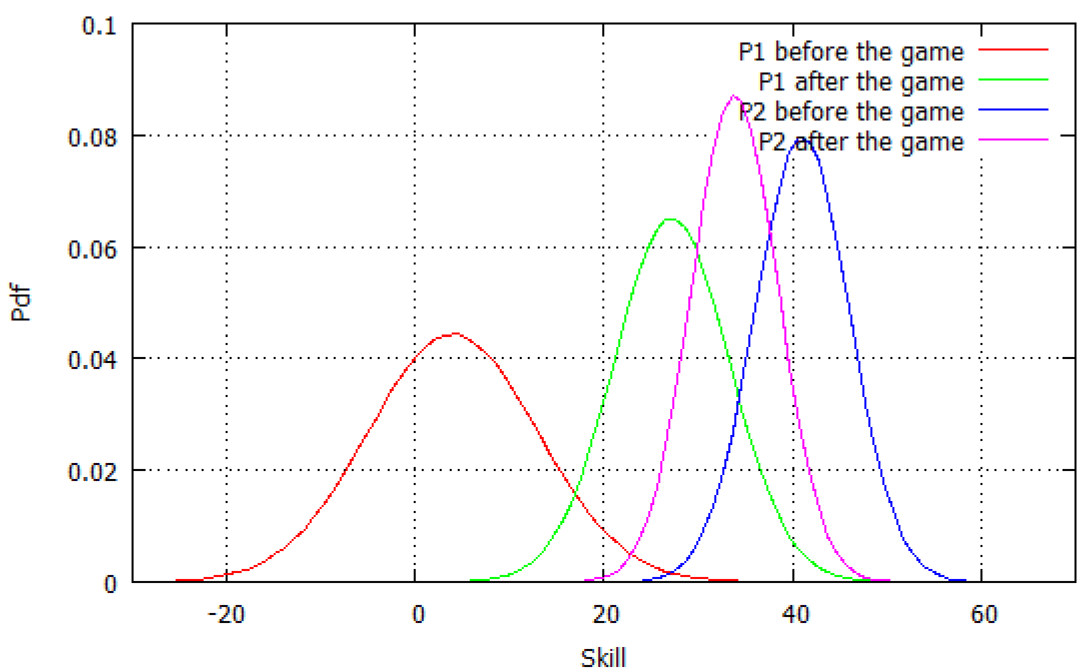

Figure 2: Skills for player p_1 and p_2 before and after the game (Player p_1 is the winner)

Bayesian Inference with Expectation Propagation

Expectation Propagation [2] is a deterministic and approximated Bayesian inference algorithm developed by Thomas Minka. It's sometimes referred as a generalization of Belief Propagation [3] algorithm, in a sense that, instead of passing exact belief messages between factors and variables in a factor graph, it sends belief expectations such as Gaussian distribution. This algorithm plays a central role in the TrueSkill rating model, by inferring the player skills and the probabilities of winning a tennis game.

As a practical example, consider the task of calculating the new value of skill for a tennis player given observed outcome of a game. We follow here the process of performing Expectation Propagation inference, presented by Thomas Minka during his lecture on Expectation Propagation that he gave at Machine Learning Summer School in Cambridge UK, 2009 [4].

First, draw a factor graph

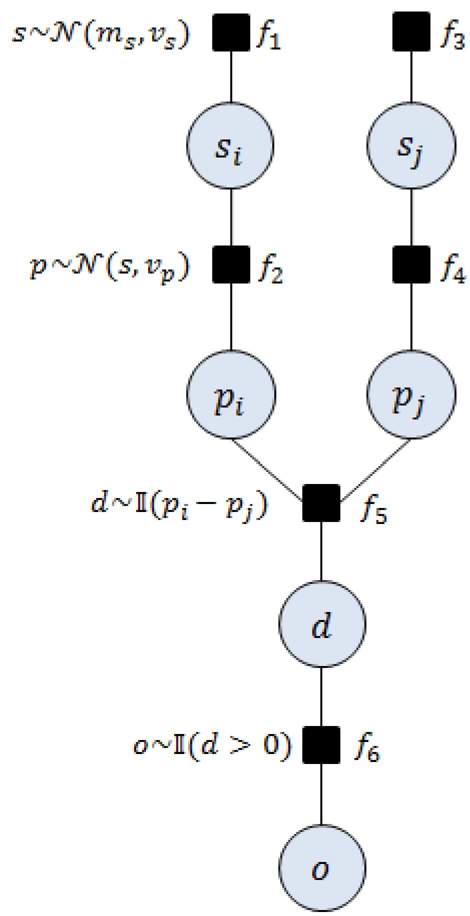

Figure 3: Factor graph for a True Skill rating model in tennis

Next, define a message schedule to be executed on a factor graph. Every message forms a uniform Gaussian distribution $\mathcal{N}(x|m, v)$. The $proj(q)$[2, 5] operation in the message $m_{f6 \to f5}$ refers to the moment matching technique for approximating function q with a Gaussian distribution.

$$m_1 : m_{f1 \to f2} \quad = \quad s_i$$

$$
\begin{aligned}
m_2 : m_{f2\to f5} &= \int m_1 p(p_i|s_i) ds_i \\
m_3 : m_{f3\to f4} &= s_j \\
m_4 : m_{f4\to f5} &= \int m_3 p(p_j|s_j) ds_j \\
m_5 : m_{f5\to f6} &= m_2 - m_4 \\
m_6 : m_{f6\to f5} &= proj(m_5 p(o|d))/m_5 \\
m_7 : m_{f5\to f2} &= m_6 + m_4 \\
m_8 : m_{f2\to f1} &= \int m_7 p(p_i|s_i) dp_i
\end{aligned}
$$

The next step involves executing the message schedule for a number of iterations till achieving some converge point. In our setup, Expectation Propagation algorithm converges after a single iteration, only because there is a single approximated message sent in a factor graph.

In the end, we can calculate the value of skill for $player_i$ by multiplying all incoming messages for the variable s_i

$$p(s_i|o) = m_{f1\to f2} m_{f2\to f1}$$

Appendix A

Example implementation of Expectation Propagation for a tennis game is available under Bayes-Scala toolbox.

References

[1] Ralf Herbrich, Tom Minka, Thore Graepel. TrueSkill TM: A Bayesian Skill Rating System, 2007

[2] Thomas P Minka. A family of algorithms for approximate Bayesian inference, 2001

[3] Christopher M. Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics), 2009

[4] Thomas Minka, Microsoft Research, Cambridge UK. Lecture on Approximate Inference. Machine Learning Summer School. Cambridge UK, 2009

[5] Daniel Korzekwa. Gaussian approximation with moment matching, aka proj() operator in Expectation Propagation, 2013

[6] Bayes-Scala Toolbox - TrueSkill, Expectation Propagation in Tennis