MiniProject Report

Anonymous Author(s)
Affiliation
Address
email

1 Q1 - The distributions and hyper-parameters

Two players are being matched against each other. Their respective skills are s_1 and s_2, such that:

$$s_1 \sim N(s_1; \mu_1, \sigma_1^2) \qquad\qquad s_2 \sim N(s_2; \mu_2, \sigma_2^2)$$

The outcome of the match is represented by a normally distributed random variable

$$t \sim N(t; s_1 - s_2, \sigma_t^2)$$

The result of the match is 1 if the outcome $t > 0$, and -1 otherwise. We call it y and,

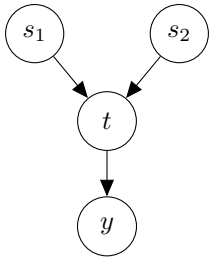
$$y = sign(t)$$

The final model is the joint-probability of all the random variables together with the set of 5 hyper-parameters.

$$p(s_1, s_2, t, y; \mu_1, \mu_2, \sigma_1, \sigma_2, \sigma_t)$$

2 Q2 - The Bayesian Network

The TrueSkill system admits the following Bayesian Network (**BN**) representation:



The paths $s_1 \to t \to y$ and $s_2 \to t \to y$ form head-to-tail paths. If t is observed then these paths are blocked. Hence,

- $\{s_1$,y$\}$ are conditionally independent on t
- $\{s_2$,y$\}$ are conditionally independent on t

3 Q3 - Some calculations using the model

3.1 Computing the conditional distribution of the skills $p(s_1, s_2|t, y)$

From Q2, we have that s_1 and y are conditionally independent on t (same for s_2). Hence,

$$p(s_1, s_2|t, y) = p(s_1, s_2|t)$$

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

Since the skills of the two players are independent, we have that

$$p(s_1, s_2) = p(s_1)p(s_2) = N\left(\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}; \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}\right) \tag{1}$$

And, by definition

$$p(t|s_1, s_2) = N(t; s_1 - s_2, \sigma_t^2) = N\left(t; (1 \quad -1)\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}, \sigma_t^2\right) \tag{2}$$

Hence, according to Corollary 1 in Lecture 2

$$\boxed{p(s_1, s_2|t) = N\left(\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}; \mu_s, \Sigma_s\right)}$$

With (all calculations made)

$$\boxed{\mu_s = \Sigma_s \begin{pmatrix} \frac{\mu_1}{\sigma_1^2} + \frac{t}{\sigma_t^2} \\ \frac{\mu_2}{\sigma_2^2} - \frac{t}{\sigma_t^2} \end{pmatrix}} \quad \boxed{\Sigma_s^{-1} = \begin{pmatrix} \frac{1}{\sigma_1^2} + \frac{1}{\sigma_t^2} & \frac{-1}{\sigma_t^2} \\ \frac{-1}{\sigma_t^2} & \frac{1}{\sigma_2^2} + \frac{1}{\sigma_t^2} \end{pmatrix}}$$

3.2 Computing the conditional distribution of the outcome $p(t|s_1, s_2, y)$

Using Bayes theorem and the conditional independence of y and s_1 then s_2 on t, we have:

$$p(t|s_1, s_2, y) \propto p(y|t).N(t; s_1 - s_2, \sigma_t^2)$$

Since y = sign(t) then

$$p(t|s_1, s_2, y) \propto \begin{cases} N(t; s_1 - s_2, \sigma_t^2) & \text{if } y.t > 0, \\ 0 & \text{otherwise} \end{cases}$$

3.3 Computing the marginal probability that Player 1 wins the game $p(y = 1)$

$$p(y = 1) = p(t > 0) = \int_0^\infty p(t)dt$$

To get p(t), we will apply Corollary 2 from Lecture 2:
From equations 1 and 2

$$p(t) = N(t; m_s, \Sigma_t)$$

With

$$m_s = (1 \quad -1)\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix} \qquad \Sigma_s = \sigma_t^2 + (1 \quad -1)\begin{pmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{pmatrix}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Conclusion:

$$\boxed{p(y = 1) = \int_0^\infty N(t; m_s, \Sigma_t)dt} \text{ With } \boxed{m_s = \mu_1 - \mu_2} \quad \boxed{\Sigma_s = \sigma_t^2 + \sigma_1^2 + \sigma_2^2}$$

4 Q4 - Gibbs Sampler

Our goal is to sample from the joint distribution $p(s_1, s_2, t|y)$ using a Gibbs sampler since $p(s_1, s_2|y)$ is intractable. After initializing s_1 and s_2 with their means, we sample t^{k+1} from $p(t|s_1^k, s_2^k, y)$ then s_1^{k+1} from $p(s_1|t^{k+1})$ and s_2^{k+1} from $p(s_2|t^{k+1})$, for k=0...L-1.

Design choices We find that it's important that we don't make any prior beliefs on the rating of the two players before any match has been played. Hence, we take the same distribution (i.e mean and variance) for s_1 and s_2. We chose $\mu_{s_1} = \mu_{s_2} = 1, \sigma^2_{s_1} = \sigma^2_{s_2} = 1, \sigma^2_t = 5$ and L=1000 samples.

4.1 Choice of the burn-in:

Figure 2 show that the distribution becomes stationary very quickly. When running the simulation with different initial conditions, it seems the burn-in is always fast. We will be discarding 10 samples to be sure that we don't include any samples from any potential burn-in period.

4.2 Approximating the distribution of the samples with a Gaussian

From the samples drawn by the Gibbs sampler, we can estimate the means and deviations $m_i = E[p(s_i|y)]$ and $\sigma_i = \sigma(p(s_i|y))$ (i$\in \{1, 2\}$). Then, we can approximate the posterior $p(s_i|y)$ with $N(s_i; m_i, \sigma^2_i)$.

4.3 Trade-off between accuracy and computational time

From Figure 3, we can see that for L=10000 samples, we obtain a very accurate estimation of the mean but the computational cost is high. 1000 samples however is not time expensive but gives a poor estimation. A good trade-off can be L=3000 samples since 5000 samples doesn't give much better results considering its computational cost.

4.4 Comparison between priors and Gaussian approximation of posteriors

As seen in Figure 4, the posterior $p(s_1|y = 1)$ has shifted to the right, while the posterior $p(s_2|y = 1)$ has shifted to the left. In fact, since we have y=1, player 1 always wins against player 2. Thus, the mean skill of player1 $E(p(s_1|y = 1))$ estimated with the Gibbs sampler should increase, while the mean skill of player2 $E(p(s_2|y = 1))$ should decrease. This observation is confirmed with Figure **??** as well.

5 Q5 - Assumed Density Filtering

Ranking results The resulting ranking and posterior skills can be seen in Table 1. In the table we can regard the posterior mean as a measure of the teams skill and order our ranking based on it. The variance can be viewed as a measure of how sure we are about the teams skill, hence a higher variance means that we are uncertain of the teams skill while a lower variance indicates that we are certain about what skill-level the team is on.

Ordering of matches affect ranking When randomizing the order in which the matches take place the ranking also changes. This is because winning/loosing a match against a team that has already lost/won a match provides a different change in the posterior. Hence the order of the matches has a direct effect on the rankings.

6 Q6 - Using the model for predictions

We suggest the following prediction algorithm: for each match, we can take the sign of the difference of the skills of the two players sampled by Gibbs+ADF, and use the result as our prediction of the next match. In other words: $y^{pred}_{k+1} = \text{sign}(s_{1|y_1..y_k} - s_{2|y_1..y_k})$. It should be noted that this however does not take in to account any information about uncertainty.

Using this predictor we reach a prediction rate of 0.625. To assess if our predictor is better than random guessing we could see if the predictor gives a result with a prediction rate that is higher than the prediction rate of random guessing (guessing with probability$= 0.5$) and if the difference statistically significant. We can do this by using a **binomial test**. If the p-value (pv) produced by the binomial test is smaller than a specified critical value ($pv < 0.001$) that means that we can reject the null hypothesis (the hypothesis that our predictor is the same as random guessing).

The p-value generated from our binomial test is ≈ 0.000045. Hence our p-value is clearly smaller than the critical value of 0.001 meaning that we are confident that our predictor produces better results than random guessing.

7 Q7 - The Factor Graph of the Model and Message-Passing

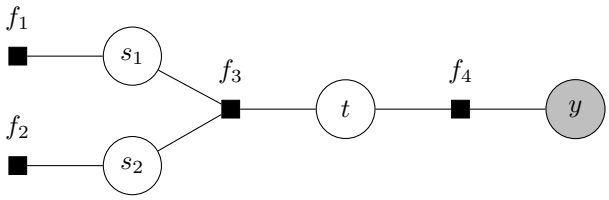


Figure 1: The factor graph of the TrueSkill model.

Expressions of factor nodes:

$$
\begin{aligned}
f_1(s_1) &= N(s_1; \mu_1, \sigma_1^2) \\
f_2(s_2) &= N(s_2; \mu_2, \sigma_2^2) \\
f_3(t, s_1, s_2) &= N(t; s_1 - s_2, \sigma_t^2) \\
f_4(t, y) &= \delta(y - sign(t))
\end{aligned}
$$

Using the message-passing algorithm one gets the following explicit messages between nodes:

Messages going to the right:

$$
\begin{aligned}
\mu_{f_1 \to s_1}(s_1) = N(s_1; \mu_1, \sigma_1^2) \quad &= \mu_{s_1 \to f_3}(s_1) \\
\mu_{f_2 \to s_2}(s_2) = N(s_2; \mu_2, \sigma_2^2) \quad &= \mu_{s_2 \to f_3}(s_2)
\end{aligned}
$$

$$
\begin{aligned}
\mu_{f_3 \to t}(t) &= \int_{s_1} \int_{s_2} N(t; s_1 - s_2, \sigma_t^2)\, \mu_{s_1 \to f_3}(s_1)\, \mu_{s_2 \to f_3}(s_2) ds_1 ds_2 = N(t;\, \mu_1 - \mu_2,\, \sigma_t^2 + \sigma_1^2 + \sigma_2^2) \\
\mu_{t \to f_4}(t) &= \mu_{f_3 \to t}(t) \\
\mu_{f_4 \to y}(y) &= \int_t \delta(y - sign(t))\, \mu_{t \to f_4}(t) dt
\end{aligned}
$$

Messages going to the left:

$$
\begin{aligned}
\mu_{y \to f_4}(y) &= \delta(y = y_{obs}) \\
\mu_{f_4 \to t}(t) &= \sum_y \delta(y - sign(t))\, \mu_{y \to f_4}(y) = \delta(y_{obs} - sign(t)) \\
\mu_{t \to f_3}(t) &= \mu_{f_4 \to t}(t) = \frac{\hat{p}(t)}{\mu_{f_3 \to t}} \propto N\left(t;\, \mu', \sigma'^2\right) \\
\mu_{f_3 \to s_1}(s_1) &= \int_{s_2} \int_t N(t; s_1 - s_2, \sigma_t^2)\, \mu_{t \to f_3}(t)\, \mu_{s_2 \to f_3}(s_2)\, dt ds_2 \\
\mu_{f_3 \to s_2}(s_2) &= \int_{s_1} \int_t N(t; s_1 - s_2, \sigma_t^2)\, \mu_{t \to f_3}(t)\, \mu_{s_1 \to f_3}(s_1)\, dt ds_1
\end{aligned}
$$

The function $\hat{p}(t)$ is a gaussian approximation via moment-matching of $p(t|y=1) \propto \mu_{f_4 \to t} \mu_{f_3 \to t}$ which would otherwise be an intractable half-gaussian. Assuming $\hat{p}(t) \sim N(t; \mu_p, \sigma_p^2)$, we can use Gaussian division to write that the message $\mu_{t \to f_3}$ is proportional to a Gaussian $N(t; \mu', \sigma'^2)$ with:

$$\mu' = \frac{\mu_p(\sigma_1^2 + \sigma_2^2 + \sigma_t^2) - (\mu_1 - \mu_2)\sigma_p^2}{\sigma_1^2 + \sigma_2^2 + \sigma_t^2 - \sigma_p^2}$$
$$\sigma'^2 = \frac{\sigma_p^2(\sigma_1^2 + \sigma_2^2 + \sigma_t^2)}{\sigma_1^2 + \sigma_2^2 + \sigma_t^2 - \sigma_p^2}$$

8 Q8 - Moment-matching and Message-passing implementation

For this section we assume that Player 1 is the observed winner, so that $y_{obs} = 1$, for simplicity, and we want to update the posterior skill level for this player. Using moment-matching we can approximate the following posterior as $\hat{p}(t)$:

$$p(t|y = 1) \propto N(t; \mu_1 - \mu_2, \sigma_t^2 + \sigma_1^2 + \sigma_2^2)\delta(t > 0)$$

Using the μ' and σ'^2 given for the Gaussian division in the previous section this results in the following message:

$$\begin{aligned}
\mu_{f_3 \to s_1}(s_1) &= \int_t \int_{s_2} N(t; s_1 - s_2, \sigma_t^2)\, N(s_2; \mu_2, \sigma_2^2)\, N(t; \mu', \sigma'^2)\, ds_2 dt \\
&= \int_t N(t; s_1 - \mu_2, \sigma_t^2 + \sigma_2^2)\, N(t; \mu', \sigma'^2) dt \\
&= N(s_1 - \mu_2; \mu', \sigma_t^2 + \sigma_2^2 + \sigma'^2) \\
&= N(s_1; \mu' + \mu_2, \sigma_t^2 + \sigma_2^2 + \sigma'^2)
\end{aligned}$$

Here Corollary 2 has been used to marginalize out s_2, then the integral of the product can be shown to give a new normal distribution which is shifted to obtain the final result.

This message gets passed along to the s_1 node and we get the approximation of the posterior seen in Figure 5. As one can see the moment-matching approximation very closely resembles the result of the Gibbs Sampling process.

9 Q9 -TrueSkill implementation on tennis data set

We decided to use data from the ATP World Tour top-tier tennis tour(1). The data set consists 747 of matches played in the tour as well as information about the players playing the matches. There are a total of 307 unique players in the data set, this means that some players only participated in as little as one single match, this of course affects our ranking since we will likely not find the players true skill. We chose to discard all the data except for the names of the players in a given match and who won the match. We then deployed the Gibbs sampler on the data set and found the ranking presented in Table 2.

We then performed one-step prediction implemented as in Q6. This produced a prediction rate of ≈ 0.57, and we again found a statistically significant difference from random guessing using the binomial test.

10 Q10 - Extension, football ranking and prediction with draws

Previously, in Q5/Q6 we ranked football teams skills using the data set `SerieA.csv`, in this when ranking the teams and when predicting the matches outcome we ignored the possibility of draws. In fact we removed draws entirely from the data set. However we want our model to take draws into consideration. Not much has to change in the Gibbs sampler to include the event of a draw, we simply decided that if there was a draw we would draw our samples for t from a normal distribution that is not truncated. For the prediction we can use the same method as previously too, but instead

of using the easy function prediction $=$ sign$(s_1 - s_2)$ we have to instead use a slightly different prediction function:

$$\text{prediction} = \begin{cases} 1, & \text{if } (s_1 - s_2) > \epsilon. \\ 0, & \text{if } -\epsilon < (s_1 - s_2) < \epsilon. \\ \text{-1}, & \text{if } (s_1 - s_2) < -\epsilon. \end{cases}$$

Where 1 indicates that team1 won, 0 indicates a draw and -1 indicates that team2 won. Here ϵ was chosen quite arbitrarily to be equal to the initial Skill prior variance, of course ϵ can be tweaked to find an optimal prediction rate through trial and error.

Using this prediction function, we get a prediction rate of ≈ 0.43, this is better than random guessing.

References

[1] Jeff Sackmann. *ATP Tennis Rankings, Results, and Stats.* https://github.com/JeffSackmann/tennis_atp, 2020. [Accessed 26 September 2020.]

Appendix

Team	Skill posterior mean	Skill posterior variance
Juventus	33.201295	0.495012
Napoli	29.089670	0.471392
Inter	14.143719	0.441789
Roma	14.080529	0.453942
Torino	13.778209	0.597514
Milan	13.299628	0.476325
Lazio	12.941589	0.428320
Sampdoria	12.320370	0.476188
Atalanta	11.498783	0.486189
Sassuolo	10.814767	0.565053
Fiorentina	10.321109	0.562804
Spal	9.822912	0.480579
Parma	8.002188	0.510393
Genoa	7.997941	0.515740
Cagliari	7.990062	0.561825
Udinese	6.888267	0.498218
Empoli	2.764128	0.480869
Bologna	0.123836	0.488804
Frosinone	-5.640880	0.556724
Chievo	-8.273269	0.582219

Table 1: Football teams from the `SerieA` data set ordered by the posterior mean of their "Skill".

Name	Skill posterior mean	Skill posterior variance
Novak Djokovic	14.890503	0.593636
Andrey Rublev	13.544298	0.571935
Gael Monfils	9.686540	0.591593
Rafael Nadal	9.330093	0.683715

Table 2: The four highest ranked players from the ATP World Tour.

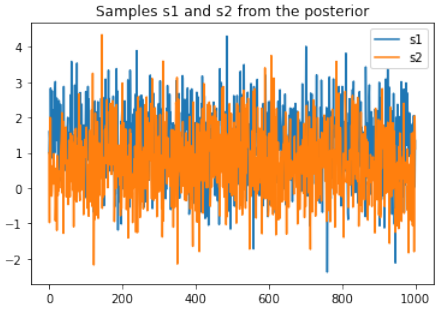


Figure 2: Samples drawn by the Gibbs sampler for L=1000: the burn-in period is not visible the distribution reaches stationarity very quickly.

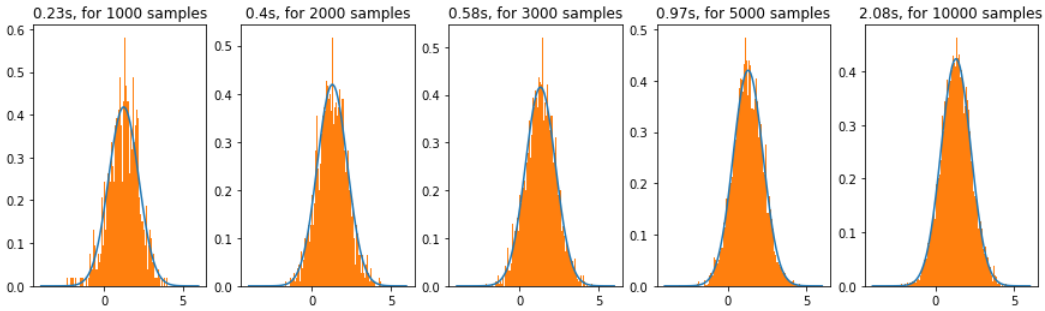


Figure 3: Analysing the effect of the number of samples on the computation time and the accuracy of the Gibbs sampler. The plots in blue represent the Gaussian approximation of the posterior $p(s_1|y=1)$. The plot in orange is the histogram of samples s_1. The same goes for s_2

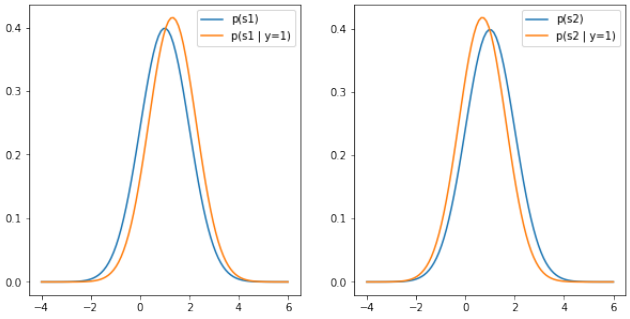


Figure 4: Comparing the prior $p(s_1)$ with $\mu_1 = 1$ and $\sigma_1^2 = 1$, to the left, (respectively $p(s_2)$ with the same distribution, to the right), with the Gaussian approximation of the posterior $p(s_1|y=1)$ (respectively $p(s_2|y=1)$) drawn from the Gibbs sampler.

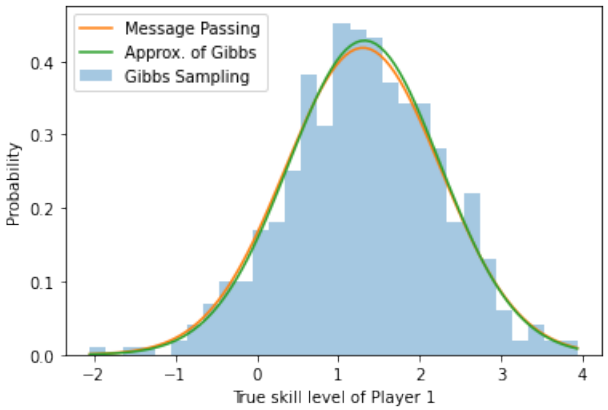


Figure 5: Two approximations of the new posterior skill level for Player 1.