Project assignment
Advanced Probabilistic Machine Learning

Oliver Groth Anton Ljungar Sofia Reineck Axel Wohlin

0 Introduction

In this project the skill of gamers in a game is estimated using a probabilistic model based on the Trueskill Bayesian ranking system developed at Microsoft Research. The ranking system is mainly used for Xbox live. However, this model can be used on many different kinds of datasets to recieve an estimate of the skill of two players. The goal in this project is to derive such a model an then use it to estimate the skills. The model is later extended taking the occurences when the two players have equal results into account, eg when both players win (or lose).

1 Q1. Modeling

Before we can proceed to any other part of this project, we need to formulate the *Trueskill* Bayesian model. To start we now that we have 4 different random variables listed below:

- 2 Gaussian random variables s_1, s_2 (skill)
- 1 Gaussian random variable t, where $\mu_t = s_1 - s_2$ (autonome)
- 1 Discrete random variable $y = \text{sign}(t)$

This gives us the following formulation of our *Trueskill* Bayesian model where we have 5 hyperparameters to define, $\mu_1, \sigma_1, \mu_2, \sigma_2$ and β^{-1}:

$$\begin{cases} p(s_1) = \mathcal{N}(\mu_1, \sigma_1) \\ p(s_2) = \mathcal{N}(\mu_2, \sigma_2) \\ p(t|s_1, s_2) = \mathcal{N}(s_1 - s_2, \beta^{-1}) \\ p(y|t) = \text{sign}(t) \\ p(s_1, s_2, t, y) = p(s_1)p(s_2)p(t|s_1, s_2)p(y|t) \end{cases} \tag{1}$$

2 Q2. Bayesian Network

In figure (1) we see the Bayesian Network for the model described in equation (1). From this network we can see that the random variables s_1 and s_2 are independent of y if t is given, since the only node between them would be given. Meaning that $s_1 \perp\!\!\!\perp y|t$ and $s_2 \perp\!\!\!\perp y|t$.

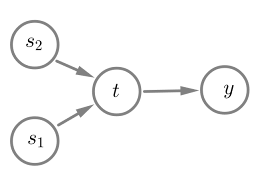

Figure 1: Bayesian Network of the model defined in equation (1)

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

3 Q3

3.1 Compute $p(s_1, s_2|t, y)$

The prior can be described as

$$p(s_1, s_2) = \mathcal{N}\left(s_1, s_2 \middle| \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}, \begin{bmatrix}\sigma_1^2 & 0 \\ 0 & \sigma_2^2\end{bmatrix}\right) \tag{2}$$

The likelihood is, as derived in equation 1

$$p(t|s_1, s_2) = \mathcal{N}\left(t \middle| X \begin{bmatrix}s_1 \\ s_2\end{bmatrix}, \beta^{-1} I_2\right), \qquad X = [1 \quad -1] \tag{3}$$

The posterior becomes

$$p(s_1, s_2|t) = \mathcal{N}(s_1, s_2|m_N, S_N) \tag{4}$$

where

$$m_N = S_N \left(\begin{bmatrix}\sigma_1^2 & 0 \\ 0 & \sigma_2^2\end{bmatrix}^{-1} \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix} + \beta\, [1 \quad -1]\, t\right), \tag{5}$$

$$S_N = \begin{bmatrix}\sigma_1^2 & 0 \\ 0 & \sigma_2^2\end{bmatrix}^{-1} + \beta \begin{bmatrix}1 \\ -1\end{bmatrix} [1 \quad -1] \tag{6}$$

3.2 Compute $p(t|s_1, s_2, y)$

From Bayes' theorem we have that

$$p(t|s_1, s_2, y) \propto p(y|t)p(t|s_1, s_2) \tag{7}$$

the conditional probability of y given t is

$$p(y|t) = \text{sign}(t) = \begin{cases} 1 & \text{if} \quad t > 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

combining this with 11 results in a truncated gaussian

$$p(t|s_1, s_2, y) \propto \begin{cases} \mathcal{N}(s_1 - s_2, \beta^{-1} I^2) & if \quad 0 < t < \inf \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

3.3 Compute $p(y = 1)$

We now want to find the marginal probability that Player 1 wins, which can be written in the following ways:

$$p(y = 1) = p(t > 0) \tag{10}$$

From before we have the conditional probability of t given s_1 and s_2, which can be seen below:

$$p(t|s_1, s_2) = \mathcal{N}\left(t \middle| X \begin{bmatrix}s_1 \\ s_2\end{bmatrix}, \beta^{-1} I_2\right), \qquad X = [1 \quad -1] \tag{11}$$

We can use the above knowledge as well as the probability functions given in the model formulation (1) to find the expected value μ_t and variance σ_t^2, used to describe the marginal probability of t given in equation (12):

$$p(t) = \mathcal{N}(\mu_t, \sigma_t) \tag{12}$$

The expected value μ_t and the variance σ_t^2 can be calculated using the equations (13) and (14) respectively:

$$\mu_t = X \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + b = [1 \quad -1] \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} + 0 = \mu_1 - \mu_2 \tag{13}$$

$$\sigma_t = \beta^{-1} + X\Sigma_{s1,s2}X^T = \beta^{-1} + [1 \quad -1] \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \beta^{-1} + \sigma_1^2 + \sigma_2^2 \tag{14}$$

The marginal probability described in (10) can be computed using the Cumulative Distribution Function (CDF) of the normal distribution in equation (12). A general definition of this CDF is described in equation (15):

$$\Phi(t; \mu, \sigma) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-(z-\mu)^2/2\sigma^2} dz \tag{15}$$

To find the probability of $t > 0$ we take the difference between the total probability, equal to 1, minus the probability of $t < 0$. The following expression can now easily be solved using commonly available software.

$$p(y = 1) = p(t > 0) = 1 - \Phi(0; \mu_t, \sigma_t) = 1 - \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{0} e^{-(z-\mu_t)^2/2\sigma_t^2} dz \tag{16}$$

4 Q4

We now want to implement a Gibbs sampler targeting the posterior distribution $p(s_1, s_2|y)$. This is done by implementing an algorithm which first generate one sample of t using equation (9) generating one sample of s_1, s_2 using the generated t and equation (4) as follows:

$$\begin{aligned} t_{\tau+1} &\sim p(t_{,\tau}|s_{1,\tau}, s_{2,\tau}, y_\tau) \\ s_{1,\tau+1}, s_{2,\tau+1} &\sim p(s_{1,\tau}, s_{2,\tau}|t_{\tau+1}, y_{,\tau}) \end{aligned} \tag{17}$$

This process is then iterated for N numbers of Gibbs iterations. To start this process we initialize $\mu_{s_1,s_2} = 1$ and $\sigma^2_{s_1,s_2} = 0$ and set $\beta^{-1} = 5$. The samples of s_1 and s_2 for 100 Gibbs iterations are plotted in figure (6). Here we see that there is a negligible burn in time, and that the variance of the samples is quite high. This could be the result of a relatively high β^{-1} value. When rerunning the experiment we see that the burn in period is still quite insignificant. We thus set the burn-in to a low value of 15, and only use the values from 16 and onwards in all the following computations. We now run the calculations for 100, 200, 600 and 1000 iterations, exclude the burn-in period, measure the execution time and calculate the mean and variance of the s_1, s_2 samples. The time, median and variance for these calculations are tabulated in table (1). In figure (2), (3), (4) and (5) we plot the histogram of the s_1 samples and plot the resulting Gaussian using the median and variance values from table (1). We see that the solution starts to converge around 1000 iterations whilst the calculation time continues to grow linearly, and thus choose this as a reasonable number of samples for this project. In figure (7) we plot the prior and posterior curves for the two players. Here we see that the posterior of the winner moves to the right and the posterior of the loser moves to the left, when compared to the prior.

Iterations	Times [s]	Skill[variance] of s_1	Median[Variance] of s_2
100	0.047	0.986[1.541]	-0.933 [1.043]
200	0.094	1.230[1.299]	-1.225 [1.046]
600	0.278	1.426[1.296]	-1.418 [1.205]
1000	0.468	1.474[1.214]	-1.437 [1.184]

Table 1: Table of the time, median and variance of the posterior of two players after 100, 200, 600 and 1000 Gibbs iterations

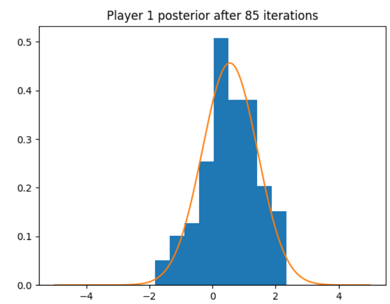

Figure 2: Player 1 posterior after 85 iterations

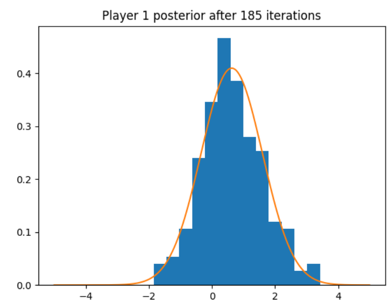

Figure 3: Player 1 posterior after 185 iterations

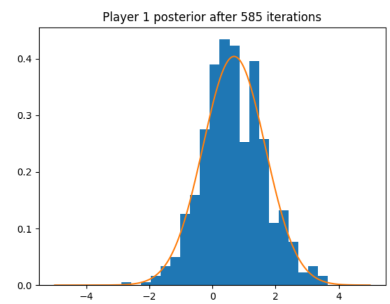

Figure 4: Player 1 posterior after 585 iterations

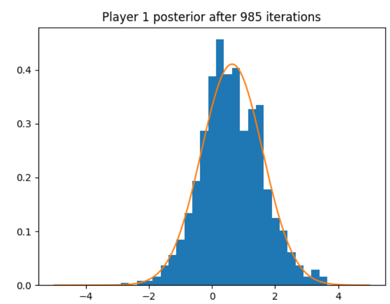

Figure 5: Player 1 posterior after 985 iterations

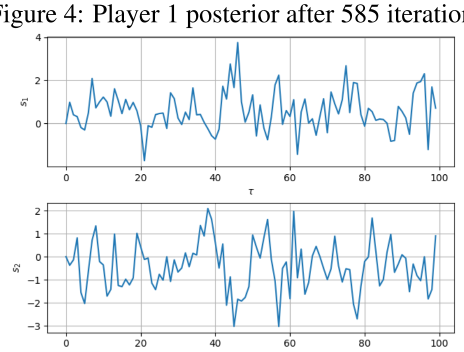

Figure 6: Plot of samples of posterior samples of s_1, s_2 for 100 Gibbs iterations

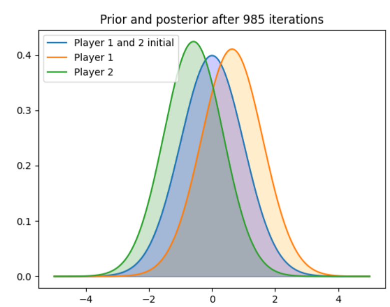

Figure 7: Prior and posterior after 9985 iterations

5 Q5

We now use the Gibbs sampler to compute the Trueskill ranking of all teams in Serie A based on match results from the 2018/19 season. This is done by using the posterior of every teams first match as the prior of their second, and then using the resulting posterior as the prior for their next match, and so on. This method of chaining priors and posteriors is called assumed density filtering (ADF) and allows us to compute the resulting skill values for arbitrarily many matches and teams. We initialize the teams skills to $\mu_{s_1,s_2} = 1$ and $\sigma^2_{s_1,s_2} = 0$, use $\beta^{-1} = 5$ and do 1000 Gibbs iterations per match. The resulting Trueskill rankings can be seen in table (2) below.

Trueskill ranking (in order)		Trueskill ranking (shuffled)		A-series Ranking
Team	Skill[variance]	Team	Skill[variance]	Team
Roma	2.68[0.17]	Napoli	2.63[0.12]	Juventus
Juventus	2.65[0.13]	Milan	2.44[0.14]	Napoli
Napoli	2.49[0.14]	Roma	1.97[0.14]	Atlantana
Atalanta	2.14[0.10]	Juventus	1.88[0.10]	Inter
Torino	2.01[0.16]	Fiorentina	1.57[0.17]	Milan
Inter	1.74[0.13]	Inter	1.36[0.15]	Roma
Milan	0.97[0.11]	Atalanta	0.83[0.13]	Torino
Sassuolo	0.08[0.15]	Lazio	0.55[0.12]	Lazio
Fiorentina	-0.07[0.14]	Torino	0.51[0.16]	Sampdoria
Lazio	-0.20[0.12]	Genoa	0.36[0.15]	Bologna
Sampdoria	-0.21[0.13]	Sassuolo	-0.52[0.15]	Sassuolo
Cagliari	-0.47[0.14]	Cagliari	-0.54[0.16]	Udinese
Udinese	-0.80[0.14]	Udinese	-0.64[0.16]	Spal
Genoa	-1.17[0.17]	Bologna	-0.85[0.16]	Parma
Spal	-1.37[0.10]	Sampdoria	-1.28[0.11]	Cagliari
Empoli	-1.44[0.10]	Frosinone	-1.62[0.13]	Fiorentina
Bologna	-1.81[0.15]	Spal	-2.21[0.18]	Genoa
Parma	-1.97[0.14]	Empoli	-2.44[0.12]	Empoli
Chievo	-2.46[0.15]	Parma	-3.31[0.16]	Frosinone
Frosinone	-2.90[0.11]	Chievo	-3.79[0.16]	Chievo

Table 2: Table of Trueskill ratings with variance for all teams in Serie A based on matches during the 2018/2019 season, calculated in order and with a shuffled data set. The final column contains their ranking according to Eurosport(1)

6 Q6

Using the Trueskill model it is also possible to predict the outcome of a match between two different players. This can be done by using the results from equation (13), (14) and (16). Here we say that if the probability $p(y = 1) > 0.5$ that player 1 wins and that if $p(y = 1)$ that player 2 wins. If we guess the result of each match based on the information from previous matches, and compute the fraction of correct guesses we can have a prediction rate in accordance with equation (18) below:

$$r = \frac{\text{number of correct guesses}}{\text{number of total guesses}} \tag{18}$$

The prediction rate for the ordered Trueskill method is compiled for various number of Gibbs iterations in table (3) below:

Number of Gibbs iterations	Prediction rate r
25	0.61
50	0.65
100	0.68
200	0.69
300	0.68
500	0.69
1000	0.70
2000	0.70
5000	0.69

Table 3: Prediction rates for various numbers of Gibbs iterations per match

7 Q7

With the factor graph we can use message passing to compute $p(t|y)$. For this we need the factor nodes

$$\begin{aligned} f_{s1}(s_1) &= \mathcal{N}(s_1; \mu_1, \sigma_1^2) \\ f_{s2}(s_2) &= \mathcal{N}(s_2; \mu_2, \sigma_2^2) \\ f_{ty}(t, y) &= \delta(y = sign(t)) \\ f_{st}(s, t) &= \mathcal{N}(t; s_1 - s_2, \beta^{-1}) \end{aligned} \tag{19}$$

The messages being passed from f_{s1} and f_{s2} are already known and can be written as

$$\begin{aligned}\mu_{f_{s1}\to s_1} &= \mathcal{N}(s_1;\mu_1,\sigma_1^2)\\ \mu_{f_{s2}\to s_2} &= \mathcal{N}(s_2;\mu_2,\sigma_2^2)\end{aligned} \tag{20}$$

We also know that the messages passed from s_1 and s_2 to f_{st} are equal to those from f_{s1} and f_{s2}.

$$\begin{aligned}\mu_{s1\to f_{st}} &= \mu_{f_{s1}\to s_1}\\ \mu_{s2\to f_{st}} &= \mu_{f_{s2}\to s_2}\end{aligned} \tag{21}$$

The last message which can be found easily is the one from y to f_{ty}.

$$\mu_{y\to f_{ty}} = \delta(y = y^*) \tag{22}$$

This can then be used to calculate the outgoing message from f_{ty}.

$$\mu_{f_{ty}\to t} = \sum_y f_{ty}(t,y)\mu_{y\to f_{ty}}(y) = \delta(y^* = sign(t)) \tag{23}$$

Continuing with the incoming and outgoing messages from f_{st}.

$$\mu_{f_{st}\to t} = \int f_{s,t}(s,t)\mu_{s\to f_{st}}(s)ds = \int \mathcal{N}(t;s_1-s_2,\beta^{-1})\mathcal{N}(s;m,S)ds = \mathcal{N}(t;m_1-m_2,\sigma_1^2+\sigma_2^2+\beta^{-1}) \tag{24}$$

The distribution $p(t|y)$ that we search for is proportional to the incoming messages $\mu_{f_{st}\to t}$ and $\mu_{f_{ty}\to t}$. The result will be a truncated Gaussian.

$$p(t|y) \propto \mu_{f_{st}\to t}\mu_{f_{ty}\to t} = \begin{cases}\mathcal{N}(t;m_1-m_2,\sigma_1^2+\sigma_2^2+\beta^{-1}), y^* = sign(t)\\ 0, else\end{cases} \tag{25}$$

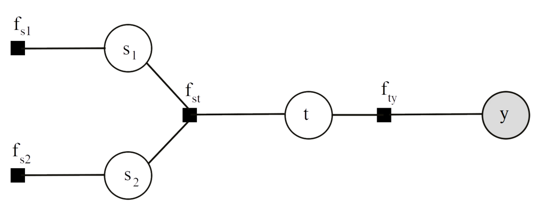

Figure 8: The factorgraph

8 Q8

We can approximate $p(t|y)$ as a gaussian function $q(t) = \mathcal{N}(t;m_3,\sigma_3^2)$. Where the parameters are calculated as:

$$m_3 = (\sigma_1^2+\sigma_2^2+\beta^{-1})\cdot\frac{\sqrt{2}}{\sqrt{\pi}}, \sigma_3 = (\sigma_1^2+\sigma_2^2+\beta^{-1})\cdot(1-\frac{2}{\pi}) \tag{26}$$

From the factor graph we can then solve the following messages

$$\mu_{t\to f_{st}} = \frac{p(t|y)}{\mu_{f_{st}\to t}} = \frac{q(t)}{\mu_{f_{st}\to t}} = \frac{\mathcal{N}(t;m_3,\sigma_3^2)}{\mathcal{N}(t;m_1-m_2,\sigma_1^2+\sigma_2^2+\beta^{-1})} = \mathcal{N}(t;m_4,\sigma_4^2) \tag{27}$$

and also the posteriors,

$$p(s1|y) = \mu_{f_{st}\to s1}\cdot\mu_{f_{s1}\to s1} = \mathcal{N}(s_1;m_2+m_4,\sigma_4^2+\sigma_1^2+\beta^{-1})\cdot\mathcal{N}(s_1;m_1,\sigma_1^2) \tag{28}$$

$$p(s2|y) = \mu_{f_{st}\to s2}\cdot\mu_{f_{s2}\to s2} = \mathcal{N}(s_2;m_1-m_4,\sigma_4^2+\sigma_2^2+\beta^{-1})\cdot\mathcal{N}(s_2;m_2,\sigma_2^2) \tag{29}$$

To calculate these we perform moment matching with respect to the initial result of y.

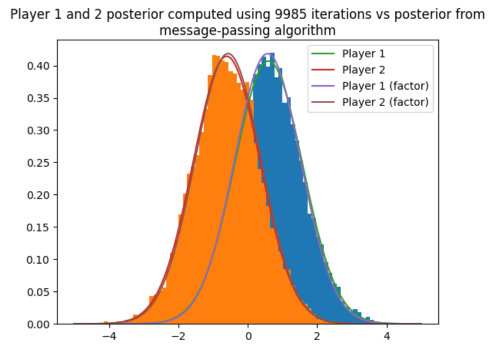

Figure 9: Player 1 and 2 posterior computed using 10000 Gibbs iterations vs posterior from message-passing algorithm

9 Q9

We now want to test our Trueskill method on a different data set. The data we chose is based on all 964 FIFA World Cup matches between 1930 and 2023(2). We initialize the teams as before to $\mu_{s_1,s_2} = 1$ and $\sigma^2_{s_1,s_2} = 0$, use $\beta^{-1} = 5$ and do 1000 Gibbs iterations per match. The resulting Trueskill rankings of the top 5 teams can be seen in table (4) below. The full table of all 82 teams can be seen in table (7) in the appendix. The dataset used contains over 40,000 different international football games ranging from friendly matches to various regional cups and world series. To reduce the computational demand of the problem we only wanted to use the most relevant matches for ranking the world's best teams. Thus all matches except for the FIFA World cup games were excluded.

Trueskill ranking (in order)	
Team	Skill[variance]
Brazil	2.76[0.03]
Italy	2.28[0.06]
Senegal	1.93[0.28]
Portugal	1.91[0.13]
Germany	1.84[0.05]

Table 4: Table of Trueskill ratings with variance for all teams participating in the FIFA world cup based on 984 matches between 1930 and 2023, calculated in order.

10 Q10

We now want to extend our model to also consider draws when calculating the skills of the teams. This is done by computing the new skill and variance assuming both a win and loss each time that the match resulted in a draw. We then take the average of the two computed new skills and variances for each team and use this as their respective new skills and variances. For the following results we initialize the teams skills to $\mu_{s_1,s_2} = 1$ and $\sigma^2_{s_1,s_2} = 0$, use $\beta^{-1} = 5$ and do 1000 Gibbs iterations per match. The prediction rate for the ordered Trueskill method including draws in its skill estimate is compiled for various number of Gibbs iterations in table (6) below. In order to be comparable to table (3) we only predict the results in games resulting in a win/loss. The resulting Trueskill rankings can be seen in table (6) below.

Number of Gibbs iterations	Prediction rate r
25	0.60
50	0.65
100	0.64
500	0.63
1000	0.67
2000	0.66

Table 5: Prediction rates for various numbers of Gibbs iterations per match when including both win/loss and draws

Trueskill ranking (with draws)		A-series Ranking
Team	Skill[variance]	Team
Juventus	1.55[0.08]	Juventus
Atalanta	1.48[0.09]	Napoli
Napoli	1.38[0.08]	Atlantana
Roma	1.28[0.10]	Inter
Torino	1.14[0.12]	Milan
Milan	0.89[0.09]	Roma
Inter	0.85[0.09]	Torino
Sassuolo	0.61[0.12]	Lazio
Lazio	-0.08[0.08]	Sampdoria
Empoli	-0.24[0.09]	Bologna
Genoa	-0.28[0.10]	Sassuolo
Cagliari	-0.37[0.08]	Udinese
Sampdoria	-0.41[0.09]	Spal
Spal	-0.58[0.09]	Parma
Fiorentina	-0.80[0.11]	Cagliari
Udinese	-0.94[0.09]	Fiorentina
Bologna	-1.21[0.11]	Genoa
Chievo	-1.29[0.12]	Empoli
Frosinone	-1.42[0.12]	Frosinone
Parma	-1.55[0.08]	Chievo

Table 6: Table of Trueskill ratings with variance for all teams in Serie A based on matches during the 2018/2019 season, calculated including both win/loss and draws. The final column contains their ranking according to Eurosport.com(1)

11 Discussion

Letting player number 1 win all the time clearly results in a higher skill in the gibbs sampling method as seen in figure 7 and 1. As discussed in the section regarding task Q4 the solution of the Gibbs sampling seem to converge after 1000 iterations because more data points are obtained and a gaussian is easier fitted to the datapoints. In table 1, the skill of player 1 seem to be around 1 and also in the plot in figure 6 the skill seem to take a random number around 1. In table 3 one can see that the predicition rate does not change significantly when using more iterations. To get a better prediction one woud have to use a more sophisticated model that takes more data into account.

Comparing the trueskill ranking in order and shuffled in table 2 the skill change a little bit. This is because in assumed density filtering the previous match results is used as posterior to the next game. It is hard from the table to determine whether it would be good to use a shuffled or in order placed data set. Considering real life conditions, that the teams play one specific game after another there could be conditions such as confidence and team building affecting the results in the next game. Therefore it could be wise to use in order data. But at the same time if we shuffle the data the Trueskill model can become more robust to variations in the order the games are played. The model becomes less sensitive to new outlying data. If the data is in order and the players have unusually strong or weak opponents in the beginning it will significantly affect the skills. Looking at table 7 where the Trueskill method was used on a different data set the result is interesting. Realistically the skill of the teams should have changed over time considering France has won several games the last few years. In the table their skill is only -0.51 whcih means that the Trueskill model would perform bad when predicting upcoming games. In this case it would be interesting to see what happended in the data was shuffled, it would probably not be very accurate since france is still considered one of the best national football teams world wide today but it might improve the predicted skill.

The model was extended in the way that the games where there was a draw between the two teams were included. This could be a more realistic approach since one team not losing but a bad team that is usually always losing doing a draw against the other team could mean that its actually better than previously seen. It would be interesting to see if the model could be improved in other ways, such as changing the model over time, looking at skills over different time periods.

References

[1] EuroSport, “Serie a table 2018/2019,” oct 2023.

[2] M. Jürisoo, “International football results from 1872 to 2023,” oct 2023.

A Trueskill ranking table of national teams (Q9)

Trueskill ranking (in order)			
Team	Skill[variance]	Team	Skill[variance]
Brazil	2.76[0.03]	France	-0.51[0.09]
Italy	2.28[0.06]	Turkey	-0.52[0.31]
Senegal	1.93[0.28]	Peru	-0.56[0.28]
Portugal	1.91[0.13]	Russia	-0.61[0.12]
Germany	1.84[0.05]	New Zealand	-0.63[0.54]
Spain	1.78[0.10]	United Arab Emirates	-0.63[0.57]
Togo	1.45[0.51]	Japan	-0.66[0.17]
Uruguay	1.44[0.09]	Panama	-0.70[0.62]
Romania	1.07[0.29]	Austria	-0.71[0.14]
Tunisia	1.03[0.22]	Ecuador	-0.79[0.28]
Yugoslavia	0.89[0.18]	Cameroon	-0.80[0.18]
Cuba	0.82[0.72]	Mexico	-0.92[0.09]
Argentina	0.69[0.04]	United States	-0.94[0.14]
El Salvador	0.41[0.44]	Slovenia	-0.98[0.51]
Iraq	0.41[0.57]	Iran	-1.03[0.24]
Greece	0.33[0.30]	Slovakia	-1.06[0.49]
Belgium	0.31[0.13]	Republic of Ireland	-1.07[0.41]
Kuwait	0.30[0.61]	Angola	-1.07[0.79]
Northern Ireland	0.24[0.31]	German DR	-1.09[0.56]
Saudi Arabia	0.22[0.21]	Qatar	-1.18[0.54]
Nigeria	0.17[0.21]	Egypt	-1.24[0.46]
South Africa	0.02[0.48]	Switzerland	-1.24[0.10]
Serbia	-0.01[0.19]	DR Congo	-1.34[0.56]
Czech Republic	-0.03[0.57]	Costa Rica	-1.42[0.20]
Hungary	-0.06[0.16]	Israel	-1.55[0.79]
Denmark	-0.07[0.21]	Colombia	-1.57[0.21]
Netherlands	-0.08[0.07]	Morocco	-1.60[0.22]
Bosnia and Herzegovina	-0.09[0.53]	Chile	-1.66[0.13]
Croatia	-0.11[0.13]	Ghana	-1.67[0.21]
Canada	-0.11[0.40]	Australia	-1.68[0.22]
Paraguay	-0.13[0.21]	Ukraine	-1.76[0.47]
Poland	-0.15[0.12]	Czechoslovakia	-1.79[0.15]
Algeria	-0.17[0.26]	Norway	-1.80[0.38]
Ivory Coast	-0.21[0.30]	Sweden	-1.97[0.11]
Jamaica	-0.22[0.62]	China PR	-2.06[0.56]
Indonesia	-0.27[0.93]	Haiti	-2.24[0.72]
Wales	-0.31[0.56]	Bulgaria	-2.28[0.18]
Trinidad and Tobago	-0.33[0.70]	South Korea	-2.30[0.09]
England	-0.36[0.08]	Scotland	-2.40[0.28]
Iceland	-0.37[0.71]	North Korea	-2.75[0.39]
Honduras	-0.42[0.36]	Bolivia	-2.93[0.60]

Table 7: Table of Trueskill ratings with variance for all teams participating in the FIFA world cup based on 984 matches between 1930 and 2023, calculated in order.