Advanced Probabilistic Machine Learning
Mini Project

Ivar Blohm, Erik Pettersson, Oskar Åsbrink, Tobias Arnehall Johansson

1 Introduction

In modern technology, ranking systems exists to match players with similar skill levels. Well known examples are Elo-rating and Trueskill, where both models uses normal distributions to match different players. Meaning, a player does not have a fixed skill, instead it is a spread based on the mean and the variance.

This project demonstrates an implementation of the Trueskill Bayesian ranking system, developed by Microsoft Research with the main purpose of ranking online matches [1]. Bayesian inference is used to find the posterior distribution of players' skill levels based on observations match results. This is however intractable and it can thus be favorable to use different approximation methods.

The Trueskill model implementation is applied to a dataset containing the results of the Italian 2018/2019 Serie A elite football division and an additional dataset containing NHL results from the same time period. Finally, improvements of the Trueskill model is presented, followed up with a discussion.

2 Assignments

Q.1 Modeling

The model consists of the following random variables and the accompanying distributions:

Variable	Description	Distribution
s_1	Skill of player 1	$p(s_1) = \mathcal{N}(s_1; \mu_1, \sigma_1^2)$
s_2	Skill of player 2	$p(s_2) = \mathcal{N}(s_2; \mu_2, \sigma_2^2)$
t	The outcome of one game	$p(t\|s_1, s_2) = \mathcal{N}(t; s_1 - s_2, \sigma_{t\|s}^2)$
y	The result of one game	$p(y\|t) = \delta(y = sign(t))$

where $\mu_1, \mu_2, \sigma_1, \sigma_2$, and $\sigma_{t|s}$ are hyperparameters. The parameters σ_1 and σ_2 represent the level of uncertainty in the players' skill level and $\sigma_{t|s}$ represents the level of uncertainty in the performance during a single game. Since s_1 and s_2 are assumed to be independent and have Gaussian distributions, the joint distribution of the variables is a multivariate Gaussian

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

which is the prior distribution in the TrueSkill Bayesian model. The full joint distribution is

$$p(s_1, s_2, t, y) = p(y|t)p(t|s_1, s_2)p(s_1)p(s_2). \tag{1}$$

Q.2 Conditional independence

Given $s = \{s_1, s_2\}$ and Eq. 1 it follows that

$$p(s, y|t) = \frac{p(s, y, t)}{p(t)} = \frac{p(y|t)p(t|s)p(s)}{p(t)} = \frac{p(y|t)p(s|t)p(s)p(t)}{p(s)p(t)} = p(y|t)p(s|t). \tag{2}$$

This implies that $s \perp y \mid t$.

Q.3 Computing with the model

3.1

Given the result in Eq. 2, we can infer that $p(s|t, y) = p(s|t)$, for which an expression can be found using *Corollary 1* [2]. Identifying $x_a = s$ and $x_b = t$ gives the following

$$p(s) = \mathcal{N}(s; \mu_s, \Sigma_s)$$
$$p(t|s) = \mathcal{N}(t; As + b, \sigma^2_{t|s})$$

where $\mu_s = \begin{bmatrix}\mu_1 \\ \mu_2\end{bmatrix}, \Sigma_s = \begin{bmatrix}\sigma_1^2 & 0 \\ 0 & \sigma_2^2\end{bmatrix}, A = \begin{bmatrix}1 \\ -1\end{bmatrix}^T s = \begin{bmatrix}s_1 \\ s_2\end{bmatrix}$ and $b = 0$.

The conditional distribution $p(x_a|x_b) = p(s|t)$ can then be written as

$$p(s|t) = \mathcal{N}(s; \mu_{s|t}, \Sigma_{s|t})$$

where

$$\mu_{s|t} = \Sigma_{s|t}(\sigma_s^{-1}\mu_s + A^T(1/\sigma^2_{t|s})(t - b))$$

and

$$\Sigma_{s|t} = (\Sigma_s^{-1} + A^T(1/\sigma^2_{t|s})A)^{-1}.$$

3.2

Moving forward, to find the full conditional distribution of the outcome $p(t|y, s_1, s_2)$, Bayes' theorem is applied as

$$p(t|s_1, s_2, y) \propto p(y|t)p(t|s_1, s_2).$$

From the model, it follows that $p(y|t) = 1$ if $y = \text{sign}(t)$ and 0 otherwise. This means that $p(t|s_1, s_2, y)$ will either be proportional to $p(t|s_1, s_2)$ or 0, denoted as

$$p(t|y, s_1, s_2) \propto \begin{cases}\mathcal{N}(t; s_1 - s_2, \sigma^2_{t|s}) & \text{if } y = \text{sign(t)} \\ 0 & \text{otherwise.}\end{cases}$$

In the case of $y = 1$ (player 1 won),

$$p(t|y, s_1, s_2) \propto \begin{cases}\mathcal{N}(t; s_1 - s_2, \sigma^2_{t|s}) & \text{for } t > 0 \\ 0 & \text{for } t < 0\end{cases}$$

and in the case of $y = -1$ (player 2 won),

$$p(t|y, s_1, s_2) \propto \begin{cases}\mathcal{N}(t; s_1 - s_2, \sigma^2_{t|s}) & \text{for } t < 0 \\ 0 & \text{for } t > 0,\end{cases}$$

which shows that $p(t|y, s_1, s_2)$ is a truncated Gaussian distribution.

3.3

To find $p(y = 1) = p(t > 0)$, s_1 and s_2 are marginalized out from the expression $p(t, s_1, s_2)$. Applying *Corollary 2* [2] with $x_a = s$ and $x_b = t$ results in

$$p(x_b) = p(t) = \mathcal{N}(t; \mu_t, \sigma_t^2) \tag{3}$$

where $\mu_t = A\mu_s + b$ and $\sigma_t^2 = \sigma_{t|s}^2 + A\Sigma_s A^T$.

Q.4 Bayesian Network

To visualize and simplify the understanding of the conditional statements in models, it is beneficial to construct Bayesian networks. They are conceptually easy to understand, where each arrow pointing from one node to another describes the dependency between the nodes (Figure 1). A gray marked node indicates that a node has been observed, which is the case in Figure 2 where t is given.

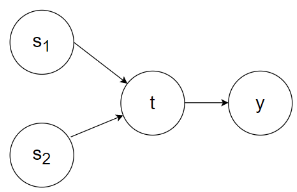

Figure 1: The Bayesian network based on the model from Q1.

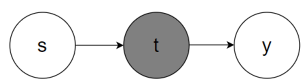

Figure 2: The Bayesian network based on the model from Q2.

Since the Bayesian Network in Figure 2 is a *head-to-tail* case (were s is the head and y the tail), the nodes are conditionally independent if and only if the node between them is observed. This concludes that $s \perp y \mid t$.

Q.5 A first Gibbs sampler

This section shows how to compute the posterior distribution of the skills s_1 and s_2 given the result of one match y based on Gibbs sampling. Using the results from Q3 it is possible to implement a Gibbs sampler that targets the posterior distribution $p(s_1, s_2 \mid y)$. Both players were given the same prior distribution of skills ($p(s_1) = p(s_2) = \mathcal{N}(\mu_0, \sigma_0^2)$) since no previous knowledge was obtained. The Gibbs sampling algorithm was performed multiple times with different hyperparameters. As no significant differences in the result were found, μ_0 and σ_0^2 were set to 0 and 1 respectively to simplify the calculation. Samples from the posterior distributions of both s_1 and s_2, with the initial condition that $y = 1$ (player 1 wins), are visualized in Figure 3.

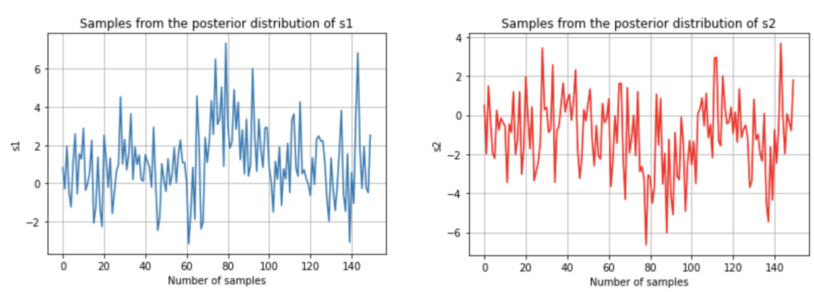

Figure 3: Samples from the posterior distributions of s_1 and s_2 generated by the Gibbs sampler when y = 1
.

From the plots in Figure 3 it is difficult to discern the so called burn-in period. This might be because the relative simplicity of the model, in combination with the fact that the initial parameter values are fairly close to the final result. However, after approximately 80 samples the probability mass tends to be above 0 for s_1 and below 0 for s_2, which is expected since player 1 won. Figure 4 shows samples from the posteriors with the burn-in period excluded. Re-running the procedure gave similar results.

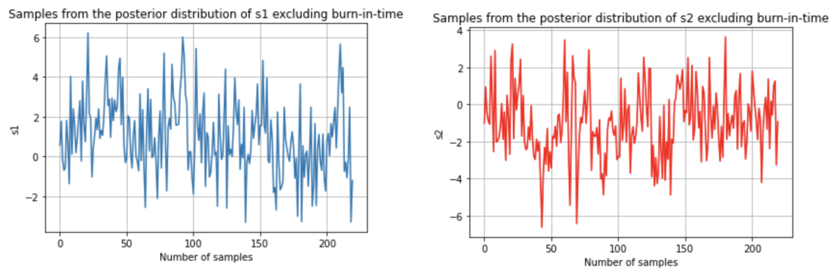

Figure 4: Samples from the posterior distributions of s_1 and s_2 generated by the Gibbs sampler when y = 1, with added burn-in value of 80.

Figure 5 shows four different plots visualizing the trade-off between accuracy of the estimate and the computational time.

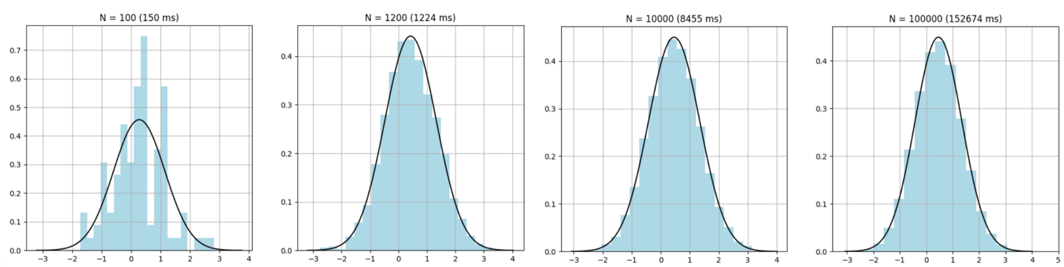

Figure 5: Histogram of the samples generated together with the fitted Gaussian posterior for different amounts of samples.

It is apparent that when $N = 10000$ the histogram is best fitted by the posterior distribution. Increasing the number of samples increases the execution time, while the results do not significantly improve. Meanwhile, decreasing the number of samples affects the results in a negative way. However, using a sample size of $N = 1200$ is time efficient, while resulting in a good fit. Therefore, this is a reasonable number of samples in this case.

In Figure 6 both prior and posterior distributions are shown together with the updated knowledge of both players respective skills.

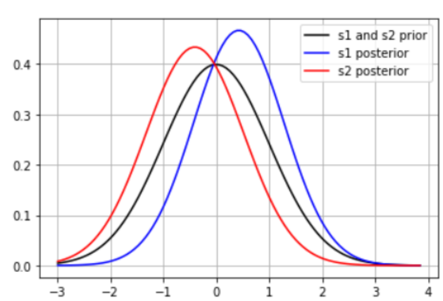

Figure 6: Comparison of prior and posterior distributions.

Both posterior distributions reflect the observed data since $p(s_1)$ has shifted to the right and $p(s_2)$ to the left. This is due to the observed value $y = 1$. Furthermore, there is a slight decrease in the variance of the posterior distributions since the observed outcome has given more information about the skill levels.

Q.6 Assumed Density Filtering

Performing ADF with Gibbs sampling to process the matches in the SerieA dataset and estimate the skill of all the teams in the dataset (shown in Figure 7), the following hyperparameters were used: $\mu_1 = \mu_2 = 0$ and $\sigma_1 = \sigma_2 = \sigma^2_{t|s} = 1$. These values were chosen for the the same reasons as given in Q5.

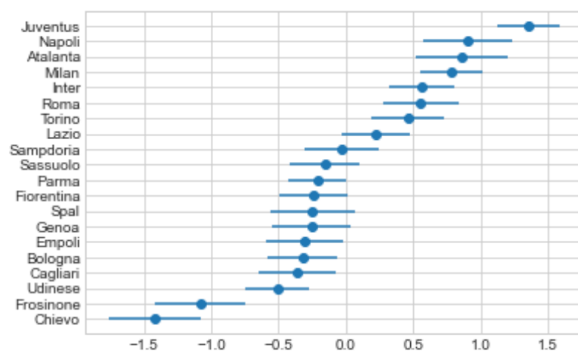

Figure 7: The skill level of each team after all matches have been processed. The blue dot represents the mean and the blue line represents the variance.

The uncertainty of the final skill levels is measured by the variance. When shuffling the order of the matches, the mean and the variance for all team changes. This is due to the fact that early results affect the skill level estimates more, since the skill level uncertainty (variance) becomes smaller and smaller during the process. Therefore, winning early and losing later on results in a higher ranking.

Q.7 Using the model for predictions

The probability that player 1 will win is equal to $P(t > 0)$. It is therefore reasonable to predict a win for player 1 if and only if $P(t > 0) > 0.5$, given the current skill estimates. Using the results for $p(t)$ from Q3, the prediction function f_{pred} can be written as

$$f_{\text{pred}} = \begin{cases} 1 & \text{if} \quad \Phi(0; \mu_t, \sigma_t^2) < 0.5 \\ -1 & \text{otherwise} \end{cases}$$

where Φ is the CDF of a normally distributed random variable. Applying f_{pred} to the dataset from SerieA results in a prediction rate of 0.64, which is clearly better than random guessing.

Q.8 Factor graph

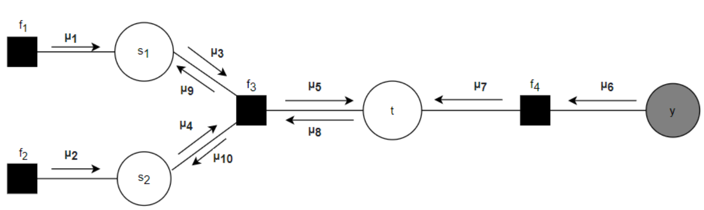

Figure 8: The factor graph of the model in Q1. Messages μ_8, μ_9 and μ_{10} relates to passing in Q9.

Explicit formula for the messages in Figure 8 are

$$\mu_1(s_1) = \mu_3(s_1) = \mathcal{N}(s_1; \mu_1, \sigma_1^2)$$
$$\mu_2(s_2) = \mu_4(s_2) = \mathcal{N}(s_2; \mu_2, \sigma_2^2)$$

denoting the mean and variance for message i as μ_i and σ_i^2. For $\mu_5(t)$ it follows that

$$\begin{aligned}\mu_5(t) &= \int_{s_1,s_2} f_3(s_1, s_2, t)\mu_3(s_1)\mu_4(s_2)ds_1ds_2 \\ &= \int_s p(t|s)p(s)ds.\end{aligned}$$

Given *Corollary 2* [2] and using Q3 to obtain

$$\mu_5(t) = \mathcal{N}(t; \mu_5, \sigma_5^2)$$

with $\mu_5 = \mu_t$ and $\sigma_5^2 = \sigma_t^2$ as in Eq 3. Furthermore

$$\mu_6(y) = \delta(y = y_{obs})$$
$$\mu_7(t) = \begin{cases}\delta(t > 0) & \text{if} \quad y_{obs} = 1 \\ \delta(t < 0) & \text{if} \quad y_{obs} = -1.\end{cases}$$

This gives for $y = 1$

$$p(t|y) \propto \mu_5(t)\mu_7(t) = \begin{cases}\mathcal{N}(t; \mu_5, \sigma_5^2) & t > 0 \\ 0 \quad \text{otherwise.}\end{cases} \tag{4}$$

For $y = -1$ the distribution for $p(t|y)$ follows from Eq. 4 trivially.

Q.9 A message-passing algorithm

The following result was obtained by using moment-matching to approximate $p(t|y)$ with a Gaussian distribution and calculate message $\mu_8(t)$:

$$\mu_8(t) = \frac{\mu_5(t)\mu_7(t)}{\mu_5(t)} \propto \frac{p(t|y)}{\mu_5(t)} \approx \frac{\hat{q}(t|y)}{\mu_5(t)} = \frac{\mathcal{N}(t; m_q, \sigma_q^2)}{\mu_5(t)} = \mathcal{N}(t; \mu_8, \sigma_8^2)$$

where

$$m_q = \frac{\sigma_5\sqrt{2}}{\sqrt{\pi}} \quad \text{and} \quad \sigma_q^2 = \sigma_5^2(1 - \frac{2}{\pi}).$$

To compute the posterior distribution for s_1 and s_2, message-passing can be used, which requires computations of $\mu_9(s_1)$ and $\mu_{10}(s_2)$. Since both messages are calculated in a similar manner (due to the symmetry of the factor graph in Figure 8), only $\mu_9(s_1)$ is derived below

$$\begin{aligned}\mu_9(s_1) &= \int_{t,s_2} f_3(s_1, s_2, t)\mu_8(t)\mu_4(s_2)ds_2dt \\ &= \int_t \mu_8(t)\left(\int_{s_2} \mathcal{N}(t; s_1 - s_2, \sigma_{t|s}^2)\mathcal{N}(s_2; \mu_2, \sigma_2^2)ds_2\right)dt.\end{aligned}$$

Using *Corollary 2* [2] on the inner integral with $x_a = s_2$, $\mu_a = \mu_2$, $\Sigma_a = \sigma_2^2$, $x_b = t$, $\Sigma_{b|a} = \sigma_{t|s}^2$, $A = -1$, and $b = s_1$, $\mu_9(t)$ can be written as

$$\mu_9(s_1) = \int \mathcal{N}(t; m_8, \sigma_8^2)\mathcal{N}(t; s_1 - \mu_2, \sigma_{t|s}^2 + \sigma_2^2)dt \tag{5}$$

$$= \int \mathcal{N}(t; m_8, \sigma_8^2)\mathcal{N}(s_1; t + \mu_2, \sigma_{t|s}^2 + \sigma_2^2)dt \tag{6}$$

where *Property 1* and *2* are used to obtain Eq. 6. Using *Corollary 2* [2] with $x_a = t$, $\mu_a = \mu_8$, $\Sigma_a = \sigma_8^2$, $x_b = s_1$, $\Sigma_{b|a} = \sigma_{t|s}^2 + \sigma_2^2$, $A = 1$, and $b = \mu_2$. This gives

$$\mu_9(s_1) = \mathcal{N}(s_1; \mu_8 + \mu_2, \sigma_{t|s}^2 + \sigma_2^2 + \sigma_8^2)$$

and

$$p(s_1|y) \propto \mu_1(s_1)\mu_9(s_1),$$

which is the posterior distribution of the skill level of player 1, given the result of one game.

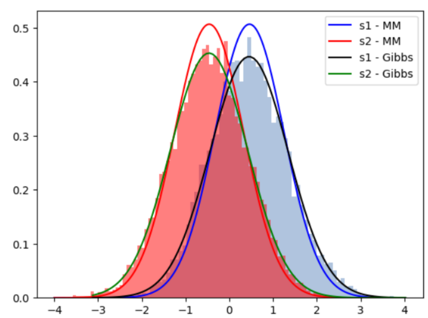

Figure 9: The posterior computed with message passing including the Gaussian approximation from Gibbs sampling and the histogram.

In Figure 9 the posteriors with message passing is visualized, as well as the responding approximation from the Gibbs sampling and the corresponding histogram from Q5. The light blue histogram corresponds to s_1, while the pink one corresponds to s_2. The posterior distribution looks rather similar for both methods, with the difference that moment matching results in less variance of the distributions. The assumption is that the Gibbs sampling procedure in itself adds some variance to the sampled values.

Q.10 Application of the model to NHL-data

The Trueskill model is not game-specific and should therefore be applicable to most (if not all) sports and competitive games. To illustrate this, the model (using Gibbs sampling) was applied to match data from National Hockey League (NHL) in the season 2018/2019 [3]. Originally the dataset contained 100 seasons of NHL matches, but was filtered out for the given period. All other pre-processing steps were conducted in a similar manner as for the Serie A dataset, including the motivation for choosing the hyperparameters. Note that there are no draws in NHL and they were therefore not taken into account. The resulting rankings are presented in Figure 10.

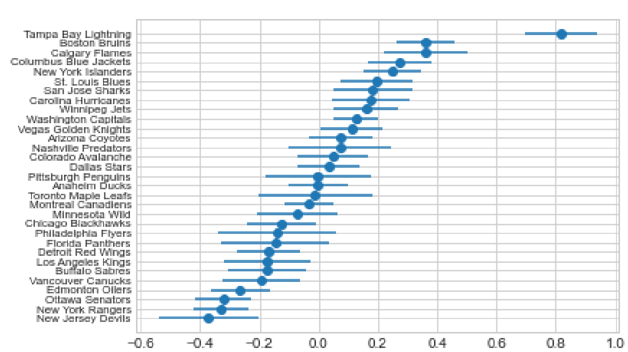

Figure 10: Teams' respective skill level after processing all matches.

The computed prediction rate was 0.53, which implies that it may be more difficult to predict NHL matches than Serie A matches. This might be due to the choice of hyperparameters, the general volatility in ice hockey and more matches played per team. Interesting to note is that Tampa Bay Lightning was ranked far better than all the other teams but did not win the season. This was because they won the group stage (in their conference), but were eliminated in the first round of the playoffs.

Q.11 Project extension - Implement draws in the model

Since there are many draws in football, any accurate model has to take them into account. To add this support to the model given in Q1, the following modifications were made: Initially, the range of the variable y was extended from $\{-1, 1\}$ to $\{-1, 0, 1\}$, with 0 representing a draw. Then p($y|t$) was defined as being 1 if ($y = 1$ and $t > \epsilon$) or ($y = 0$ and $|t| < \epsilon$) or ($y = -1$ and $t < -\epsilon$), and 0 otherwise. Here, ϵ is a new hyperparameter indicating the range of t that should be associated with draws. The new distribution of $y|t$ results in the following expressions for $p(t|y, s_1, s_2)$.

In the case of $y = 1$ (player 1 won),

$$p(t|y, s_1, s_2) \propto \begin{cases} \mathcal{N}(t; s_1 - s_2, \sigma^2_{t|s}) & \text{for } t > \epsilon \\ 0 & \text{for } t < \epsilon \end{cases}$$

in the case of $y = -1$ (player 2 won),

$$p(t|y, s_1, s_2) \propto \begin{cases} \mathcal{N}(t; s_1 - s_2, \sigma^2_{t|s}) & \text{for } t < -\epsilon \\ 0 & \text{for } t > -\epsilon, \end{cases}$$

and for the case of $y = 0$ (draw),

$$p(t|y, s_1, s_2) \propto \begin{cases} \mathcal{N}(t; s_1 - s_2, \sigma^2_{t|s}) & \text{for } |t| < \epsilon \\ 0 & \text{for } |t| > \epsilon, \end{cases}$$

which shows that $p(t|y, s_1, s_2)$ is still a truncated Gaussian distribution, but with different limits than in the model without draws. The prediction function from Q7 was modified as to predict the result with the largest probability mass out of the three possible alternatives. The probabilities ($P(t > \epsilon)$, $P(t < -\epsilon)$ and $P(|t| < \epsilon)$) were calculated using the CDF for $p(t)$, in a similar way as in Q7.

The extended version of the model was applied to the dataset from Serie A. Out of the tested ϵ values, $\epsilon = 0.25$ was found to be optimal, resulting in a prediction rate of 0.51. The result is significantly better than random guessing, which in this case would result in a prediction rate of $1/3$.

3 Discussion

In this project, a Trueskill Bayesian algorithm was implemented to estimate the skills of teams in two different sports. Also, two different approximation methods were used, Gibbs sampling and message passing. The result shows that both methods produces a similar outcome, as expected. Worth to mention is that Gibbs sampling is more computationally expensive, while message passing requires finding expressions for the messages involved. In general, this can be quite complicated. Also, one extension to the Trueskill model was implemented, where draws were taken into account.

For further investigation of the practical use of Trueskill one could extend the model to several players, something that would enable the analysis of golf player skills for instance. Also, it would be intersting to estimate the upcoming or unfinished season in any football league. This would be a valid test of the quality of the model.

References

[1] Herbrich, Minka, Graepel. Trueskill(TM): A Bayesian Skill Ranking System, 2007.

[2] Formula sheet for the gaussian distribution advanced probabilistic machine learning 2021. https://uppsala.instructure.com/courses/71173/files/3678409?wrap=1. Accessed: 2022-09-30.

[3] skillalytics nhl data. https://www.skillalytics.com/data/nhl/. Accessed: 2022-09-30.