Advance Probablistic Machine Learning Mini Project

Anonymous Author(s)
Affiliation
Address
email

Abstract

The main goal of the project is to implement a Bayesian method, based on the TrueSkill ranking system to predict the match results between teams with the help of the previous matches scoring data. In this report, a comparison between two approximation methodologies will be conducted as well as a proposal for the Trueskill model enhancement.

Number of group members: 3.

1 Introduction

The main purpose of the project is to explore the Trueskill framework designed for multi-player games. The ranking system purpose is to identify and track the skills of players from previous games moreover build a prediction model that relies on these data.
Initially, Bayesian model is defined with its random variables. This model will use the skills difference of the players and try to rank them using Trueskill Bayesian positioning framework created by Microsoft Research for raking online matches. The ability to scientifically rate players in sports and games is extremely useful in-game matching, which helps to produce entertaining and even gaming experiences, as well as for deciding tournament qualification. It enables us to look at individual skill development which we can use to make comparisons between players and teams.

1.1 Trueskill

It is a skill-based positioning framework created by Microsoft for utilizing video game matchmaking on Xbox Live. This is specially designed for games involving more than two players, like football, etc. The reason for a positioning framework is to both distinguish and track the abilities of gamers in a game in order to be able to coordinate them into competitive matches. TrueSkill has been utilized to rank and coordinate players in numerous diverse games. Trueskill uses Bayesian approach to find the skills of a player given the past data of the games.[5].

1.2 Model

As per the given data, the model have four random variables as given below, s_1 and s_2 are the skills of the players and Gaussian random variables, another Gaussian random variable t with mean equal to s_1 - s_2 and y=sign(t), a discrete random variable for the output.

$$p(s_1) = N(s_1; \mu_{s_1}, \Sigma_{s_1}), \quad p(s_2) = N(s_2; \mu_{s_2}, \Sigma_{s_2}) \tag{1}$$

$$p(t|s) = N(t; s_1 - s_2, \Sigma_{t|s}) \tag{2}$$

$$p(y|t) = \begin{cases} 1, & \text{if } y = \text{sign}(t) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

s_1 and s_2 are independent Gaussian random variables, while t depends on s_1 and s_2 and y also depends on t so, our final model becomes as follows.

$$p(s_1, s_2, t, y) = p(s_1)p(s_2)p(t|s_1, s_2)p(y|t) \tag{4}$$

1.3 Conditional independence

Let assume that the first two random variables are s and y, as per our model and the third discrete random variable is t. Given that:

$$p(s, y|t) = \frac{p(s, y, t)}{P(t)}, \quad p(s, y|t) = \frac{p(s)p(t|s)p(y|t)}{P(t)} \tag{5}$$

where,

$$p(t|s) = \frac{p(s|t)P(t)}{p(s)} \tag{6}$$

by putting eq 6 into eq 5 we get,

$$p(s, y|t) = p(s|t)p(y|t) \tag{7}$$

so,

$$s \perp y|t \tag{8}$$

1.4 Computing with the model

1.4.1 The full conditional distribution of the skills

The goal is to compute $P(s_1, s_2|t, y)$.As approved before s_1 and s_2 are independent of y given t, so the conditional distribution of s_1 and s_2 becomes $P(s_1, s_2|t)$. Considering that P(s) and P(t|s) are known, Gauss Corollary1 (Affine transformation – conditional) can be used in order to obtain $P(s_1, s_2|t, y)$. Given that,

$$P(s) = N(s; \mu_s, \Sigma_s) \tag{9}$$

$$P(t|s) = N(t; As + b, \Sigma_{t|s}) \tag{10}$$

Then the conditional distribution of ’s’ given ’t’ is

$$p(s|t) = N(s; \mu_{s|t}, \Sigma_{s|t}) \tag{11}$$

with

$$\begin{aligned} \Sigma_{s|t} &= (\Sigma_s^{-1} + A^T \Sigma_{t|s}^{-1} A)^{-1} \\ \mu_{s|t} &= \Sigma_{s|t}(\Sigma_s^{-1}\mu_s + A^T \Sigma_{t|s}^{-1}(t - b)) \end{aligned} \tag{12}$$

where

$$S = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}, A = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad \mu_s = \begin{bmatrix} \mu s1 \\ \mu s2 \end{bmatrix}, \quad \Sigma_s = \begin{bmatrix} \sigma_{s_1}^2 & 0 \\ 0 & \sigma_{s_2}^2 \end{bmatrix} \tag{13}$$

1.4.2 The full conditional distribution of the outcome

To compute $P(t|s_1, s_2, y)$, It’s known that:

$$p(t|s_1, s_2, y) = p(y|t)p(t|s_1, s_2) \tag{14}$$

where

$$p(y|t) = \begin{cases} 1, & \text{if } y = \text{sign}(t) \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

$$p(t|s_1, s_2) = N(t; s_1 - s_2, \sigma_t^2) \tag{16}$$

The conditional distribution $P(t|s_1, s_2, y)$ is Truncate Gauss as it is multiplication of Gauss function and sign function if t range from $(0, \infty)$.

1.4.3 The marginal distribution of the outcome

In order to find p(y=1)=p(t>0). First the value of p(t>0) should be computed . Refering to eq(11) and eq(12) and applying Gauss Corollary 2. Then:

$$\begin{aligned} p(t) &= N(t; \mu_t, \Sigma_t) \\ \mu_t &= A\mu_s + b \\ \Sigma_t &= \Sigma_{t|s} + A\Sigma_s A^T \end{aligned} \tag{17}$$

Then by computing

$$P(t > 0) = \int_0^\infty P(t)dt \tag{18}$$

we can get the value of $p(t > 1)$

1.5 Bayesian Network

Bayesian network is a way of representing the probability dependencies between events(variables) using the graphical theory in order to get detailed inference of the model. In Trueskill the skill for player 1 is represented as a separate node from the skill of player2 , adding another node to represent t (the outcome of the game), Finally node for y for the game result. Now the relationships and dependencies between these variables is represented using arrows . It's obvious that the result of the game depends on the outcome of the game , as well as the outcome of the game is based on the players skills . The following graph model in Figure 1 is Bayesian Network representation of our Trueskill model. Using the Bayesian Network rules, the conditional independence between S and y as $s \perp y|t$ can be approved.

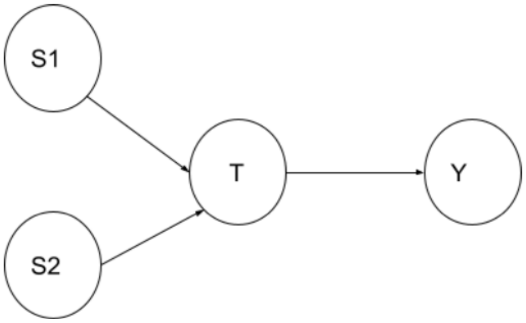


Figure 1: Bayesian Network for Trueskill model

In our case we will use Head to Tail node rule[6] giving t is observed.

$$P(s, y|t) = \frac{p(s, y, t)}{p(t)} \tag{19}$$

by replacing the join probability p(s,y,t)we get

$$P(s, y|t) = \frac{p(y|t)p(t|s)p(s)}{p(t)} \tag{20}$$

From Bayesian equation We can replace

$$\frac{p(t|s)p(s)}{p(t)} = p(s|t) \tag{21}$$

Then our final equation will be

$$P(s, y|t) = p(y|t)p(s|t) \tag{22}$$

So, S and y are conditional independent given t, $s \perp y|t$ [6].

1.6 Gibbs Sampling

In statistics, sometimes, it is hard to know joint distribution explicitly, instead, the conditional distribution of each variable is known. In this case, to get draws from joint distribution, more specifically to approximate joint distribution, Gibbs sampling is utilized [1].

Here, the method will be applied to find p(s_1,s_2| y). The samples of the posteriors generated by Gibbs sampling when y = 1 is shown in Figure 2. In this part, 5 hyper-parameters were used to implement Gibbs sampling. These hyper-parameters are μ_{s_1,s_2} = 25, σ_{s_1,s_2} = 25/3, and σ_t = 25/6. The two given conditional distribution, p(t | s_1,s_2,y) and p(s_1,s_2| t,y) are used for sampling. Initially, s_1 and s_2 values are achieved from model parameters. By using the values of s_1 and s_2, get the sample from p(t | s_1,s_2,y) and then use that sample to take a sample from p(s_1,s_2| t,y). This gives new values for s_1 and s_2.

1.6.1 Burn-in period

The initial samples are strongly reliant on the starting conditions,as a result, Gibbs sampling will not generate meaningful statistics until convergence is achieved. The phase preceding convergence is referred to as burn-in. After multiple iterations, the burn-in time interval appeared in the first 15 samples only until reaching the stationary distribution. This was achieved by obtaining variables values from the official website. [4].

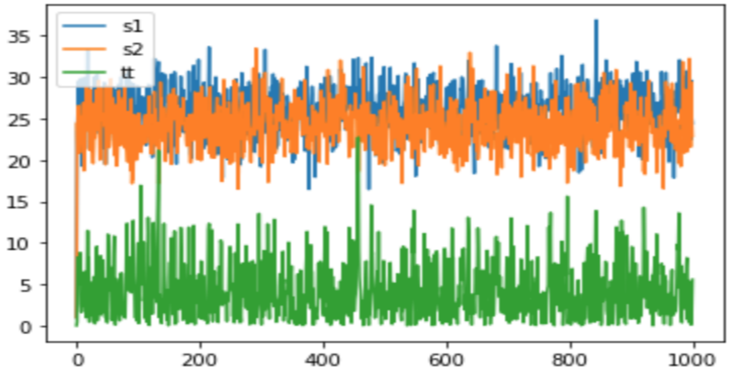


Figure 2: This figure shows the samples of posteriors and the outcome when y is 1.

1.6.2 Fitted Gaussian Posterior

As the number of samples increases, the samples become closer and closer to the fitted Normal posterior, as intended. Figure 3 shows that 10 samples may not be adequate, but 500 samples as shown in figure 4 result in a very attractive "Gaussian shape" of the sample distribution with the running time of 0.322 sec. When the number of samples is doubled, the running time is doubled. Raising the number of samples results in a closer fit, but because the fitted curve is merely an estimate, 500 samples should suffice for our needs. When compared to the computing cost, increasing the number of samples to 1000 and 10000 yields relatively little benefit, shown in figure 5 and figure 6 respectively.

Gaussian approximation of the skills after the posterior distribution can be seen in Figure 7. According to the Figure, the mean of s_1 is greater than that of s_2, indicating that s1 should be the winner because we indicated the y=1.

1.7 Assumed Density Filtering

After using Gibbs sampler in order to generate samples for the posterior p(s | t). We need to get closer to the true approximation by updating the value of posterior with more observations. Reaching this goal can be done by using the latest posterior as a prior for the next iteration[3]. By applying ADF on our Series A data-set we got skill range from 33.04 to 22.61 and standard deviation range from 1.56 to 1.32. "Juventus" team is ranked first place and by comparing it with the real data we found it’s true

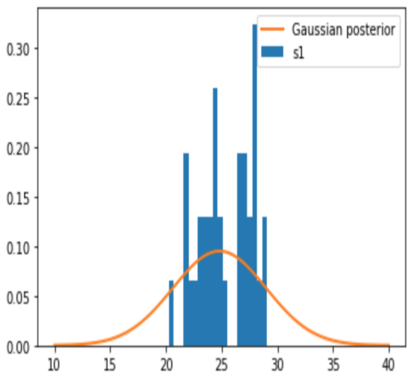


Figure 3: This figure shows the Gaussian approximation of the skills in terms of histogram visualization and curve visualization when the sample size equals to 10.

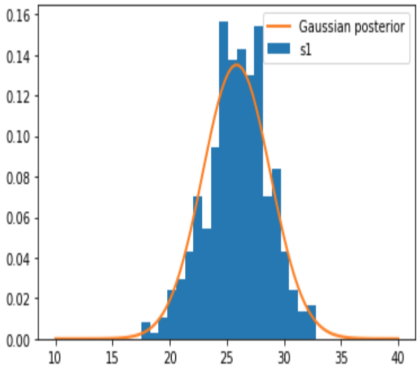


Figure 4: This figure shows the Gaussian approximation of the skills in terms of histogram visualization and curve visualization when the sample size equals to 500.

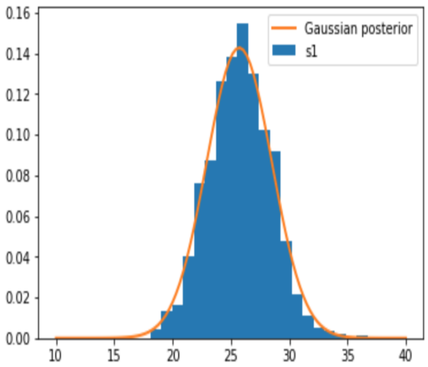


Figure 5: This figure shows the Gaussian approximation of the skills in terms of histogram visualization and curve visualization when the sample size equals to 1,000.

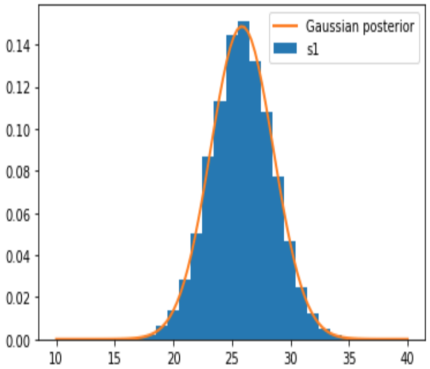


Figure 6: This figure shows the Gaussian approximation of the skills in terms of histogram visualization and curve visualization when the sample size equals to 10,000.

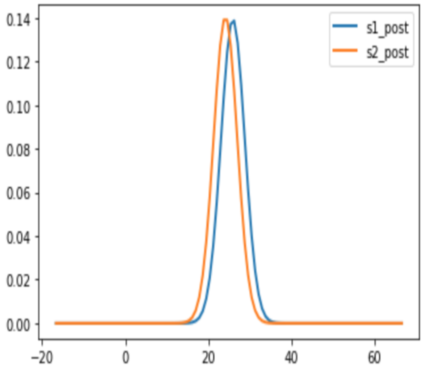
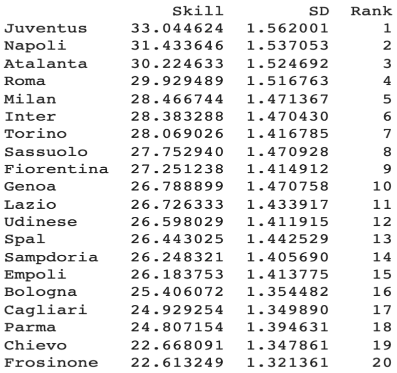


Figure 7: This figure shows the Gaussian approximation of the skills in terms of histogram visualization and curve visualization when the sample size equals to 500.

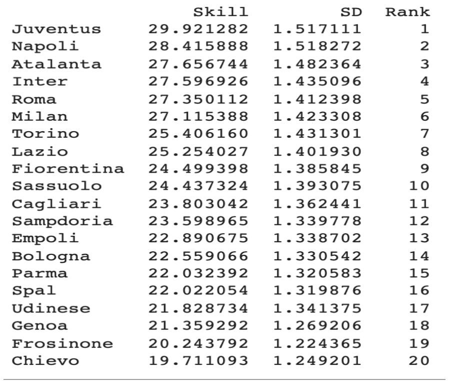
"Juventus" is the champion for season 2018/2019 and the top teams in our results fall in the same group in the real data. The final result table for the whole teams is in Figure 8.

Variance(standard deviation to the power 2) here is representing how certain we are about the skills of the team. When we change the order of the matches at random the team's skills doesn't changed significantly specially for the teams with higher skills , The randomness may affect the skills for individual players but overall the teams ranking still the same except for few teams. Results shown in Figure 9.



```
                Skill        SD  Rank
Juventus    33.044624  1.562001     1
Napoli      31.433646  1.537053     2
Atalanta    30.224633  1.524692     3
Roma        29.929489  1.516763     4
Milan       28.466744  1.471367     5
Inter       28.383288  1.470430     6
Torino      28.069026  1.416785     7
Sassuolo    27.752940  1.470928     8
Fiorentina  27.251238  1.414912     9
Genoa       26.788899  1.470758    10
Lazio       26.726333  1.433917    11
Udinese     26.598029  1.411915    12
Spal        26.443025  1.442529    13
Sampdoria   26.248321  1.405690    14
Empoli      26.183753  1.413775    15
Bologna     25.406072  1.354482    16
Cagliari    24.929254  1.349890    17
Parma       24.807154  1.394631    18
Chievo      22.668091  1.347861    19
Frosinone   22.613249  1.321361    20
```

Figure 8: The teams ranking result using Assumed Density Filtering.



```
                Skill        SD  Rank
Juventus    29.921282  1.517111     1
Napoli      28.415888  1.518272     2
Atalanta    27.656744  1.482364     3
Inter       27.596926  1.435096     4
Roma        27.350112  1.412398     5
Milan       27.115388  1.423308     6
Torino      25.406160  1.431301     7
Lazio       25.254027  1.401930     8
Fiorentina  24.499398  1.385845     9
Sassuolo    24.437324  1.393075    10
Cagliari    23.803042  1.362441    11
Sampdoria   23.598965  1.339778    12
Empoli      22.890675  1.338702    13
Bologna     22.559066  1.330542    14
Parma       22.032392  1.320583    15
Spal        22.022054  1.319876    16
Udinese     21.828734  1.341375    17
Genoa       21.359292  1.269206    18
Frosinone   20.243792  1.224365    19
Chievo      19.711093  1.249201    20
```

Figure 9: The teams ranking result After shuffling the data.

1.8 Using the model for predictions

The model follows the one-step-ahead approach and try to predict the match results by using predefined model of Skills normal distribution , t which is the mean difference between scores and finally y is calculated by y=sign(t) and the predicted output is then compared with the true results. With the help of ADF and Gibbs sampling, the model is updated which will be used again in prediction for next iterations.Prediction rate was calculated by

$$r = \frac{numberOfTruePrediction}{totalnumberofpredictions} \tag{23}$$

and obtained results were as following:
The true rate with draws = 45.526315789473685 The true rate without draws = 63.60294117647059

As per the result, it is noted that the prediction rate increase by increasing the number of iterations and obviously it's better than random guessing.

1.9 Factor Graph

To factorize the distribution function visually we can use factor graphs. These graphs works much better than Bayesian graph to solve inference issues. These graph are bipartite, means made by two different parties, variable nodes and factor nodes [7]. Variable nodes are visualized as circle and factor nodes as squares in the graph and are represented as x[i] and f[i], respectively. In factor graphs the edges are between two different parties that are factor nodes and variables nodes and these shows the dependencies of factors on variables. The factor graph's structure provides data about the variables conditions and serves as the foundation for a competent inference algorithm. The factor graph is shown in figure 10.

$$\begin{gathered} fa = N(s_1, \mu_1, \sigma_1^2), \quad fb = N(s_2, \mu_2, \sigma_2^2) \\ fc = N(t, \mu_3, \sigma_3^2)), \quad fd = N(y = sign(t)) \end{gathered} \tag{24}$$

$$\mu_y = 1, \quad \mu_f d = \sum fd\mu_y \tag{25}$$

$$\begin{gathered} \mu_f a = fa, \quad \mu_f b = fab \\ \mu_s 1 = \mu_f a, \quad \mu_s 2 = \mu_f b \end{gathered} \tag{26}$$

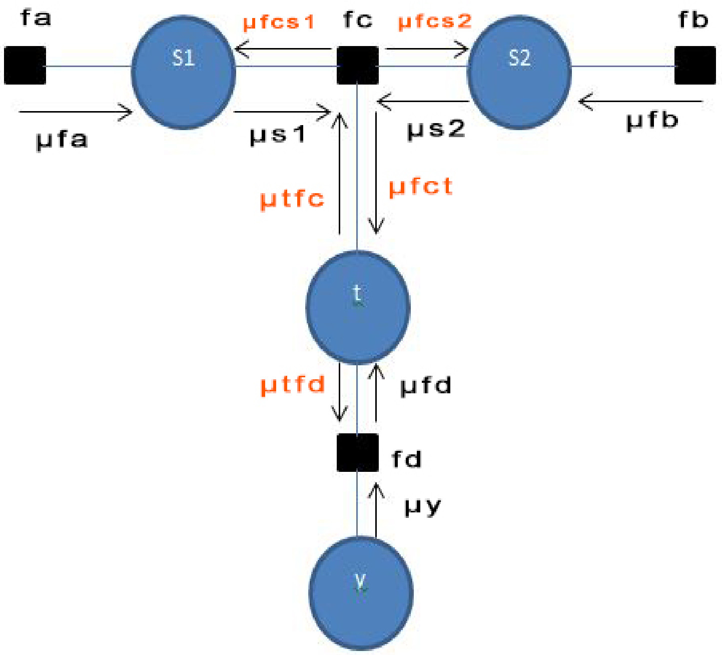


Figure 10: Showing a factor graph with messages.

$$\begin{aligned} \mu_f ct &= \iint fc\mu_{s_1}\mu_{s_2} ds1ds2 \\ \mu_t fc &= \iint P(t|s_1, s_2)P(s1)P(s_2)ds1ds2 \end{aligned} \tag{27}$$

Using corollary 2 we get:

$$\mu_{fct} = N(t; \mu_1 - \mu_2, \sigma_t^2 + \sigma_1^2 + \sigma_2^2) \tag{28}$$

where,

$$P(t|y = 1) \propto (\mu_f ct)(\mu_f d) \tag{29}$$

By putting the required values in eq 29, required result for P(t | y=1) has been achieved.

1.10 Message Passing

The below graphs shown in figure 11 and figure 12 shows the posterior distributions for s_1 and s_2 for both message passing and Gaussian approximation given the result of the one match. By looking at these graphs, it is concluded that the posterior distributions for both methods are quite similar.

1.11 Test Model on different data-set

As a part of applying the Trueskill model on different sports, The model was applied to the hockey games (National Hockey League season 2018/2019) dataset[2]. It is quite similar to our "SerieA.csv" but there is a slight difference between hockey game football matches. In the draw situation, the match goes to extra time and if the tie is not broken in extra time, the penalty shootout determines the winner. As a result, the "hockey.csv" dataset has no draw cases. A Slightly data processing was needed in order to get the same data structure obtained by the "SeriesA" dataset like changing the headers labeling. After that, we used our model in order to predict the winning teams. In the end, the ranking of teams, as well as the rate of true prediction, were computed. Our model got a prediction

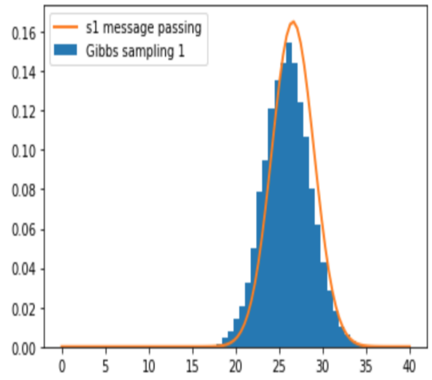


Figure 11: Message passing plotted together with Gaussian approximation of posterior for s1

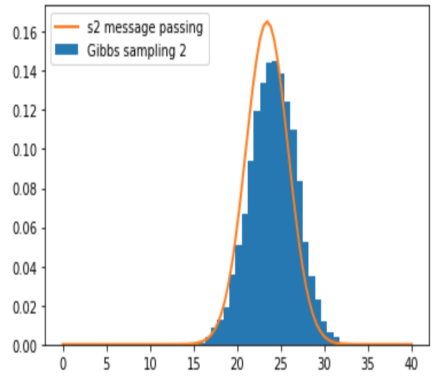


Figure 12: Message passing plotted together with Gaussian approximation of posterior for s2

rate of 53.10%. However, A better result was expected than the one for SerieA because there are no draw cases in the Hockey datasets. One of the reasons might be that the Hockey game has a lot of team players changing during the game.

1.12 Extension

In this part, it is figured out that 108 in 380 games ended with a tie, which means approximately every 1 game ended up with a draw in 3 games. For this reason, some extra features were added to the 'prediction' function in order to include the tie scenario. Firstly, a random number is created ranging from [0,2] for each iteration. Then, the modular of the iteration number and the random number is calculated. If this modular equals 0 and the absolute value of t is smaller than 0.1, we assume that this game was a tie. Then the ADF modeling was conducted as before. Our model accuracy was found out to be 47.89%. This result showed that our implementation was not good for the overall dataset because we got less accuracy from previous tasks. However, when we excluded draw games in the SerieA datasets, the accuracy, compared with the previous task, was increased to 66.91% , which means that our model determined non-tie games better than the other models. Those accuracy values were found after 3 tests and those values were the greatest ones.

1.13 Conclusion

By comparing Gibbs Sampling and Message Passing, we discovered that, while the two strategies perform similarly, both have advantages and disadvantages. Gibbs Sampling is simple to build and use for inference, but it is costly in terms of computing. Message Passing, on the other hand, uses a more efficient method. However, as the number of variables in the Bayesian Network rises, so does the complexity.Modeling for hockey matches was used to construct the ranking techniques in this study. Given a big enough number of matches, the model's prediction accuracy is reasonable. It would most likely work on related sports, such as basketball and football.
Microsoft has released an updated version of the algorithm, named as TrueSkill2 [5], that allegedly has considerably greater prediction accuracy but is more complex and customized for their needs.

References

[1] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. Part i fundamentals of bayesian inference. *Bayesian Data Analysis*, 3:4–29, 2014.

[2] hockey referenc. Hockey dataset 2018/2019.

[3] Heejin Jeong, Clark Zhang, George J Pappas, and Daniel D Lee. Assumed density filtering q-learning. *arXiv preprint arXiv:1712.03333*, 2017.

[4] Heungsub Lee. Trueskill the video game rating system.

[5] Tom Minka, Ryan Cleven, and Yordan Zaykov. Trueskill 2: An improved bayesian skill rating system. *Tech. Rep.*, 2018.

[6] Antônio Horta Ribeiro. *Bayesian Graphical Models*. 2021.

[7] Niklas Wahlström. *Message passing*. 2021.