Ranking of NBA teams using TrueSkill, a Bayesian Skill Ranking System.

Treball Fi de Grau de
Aleix Pujol i Quintana

Director: Vicenç Gómez i Cerdà

Grau en Enginyeria Matemàtica en Ciència de les Dades

Universitat Pompeu Fabra

Curs 2020-2021

Dedicat als meus pares, per les oportunitats
i l'educació que m'han donat.

1 Abstract

1.1 English version

In this research, the TrueSkill model [8] is introduced and tested to compute skills and perform predictive analysis in both the players and teams of the NBA basketball league. This model, developed by Microsoft Research, is used for ranking and player matchmaking on Xbox LIVE, and has been proved to consistently work on other sports such as tennis, golf or chess. TrueSkill is a Bayesian approximate inference algorithm, which provides a rating system for players involved in a game. TrueSkill takes advantage of message passing (Sum-Product algorithm) and expectation propagation for message approximation techniques in factor graphs, to construct the model to infer teams and players skills. In this research, different implementations for TrueSkill model are analysed and evaluated using data from the 2019/2020 regular season. Later, such inferred skills are used to perform predictive analysis on every game in the 2020/2021 regular season. Results for both inferred skills and game outcome predictions are discussed and compared with ground truth rankings using Kendall Tau distance and correlation coefficient metric measures.

1.2 Catalan version

Aquest projecte introdueix I testeja el model de TrueSkill, amb l'objectiu de calcular les habilitats i desenvolupar un model predictiu tant a nivell de jugadors com a nivell d'equips de la lliga professional de bàsquet

NBA. TrueSkill, desenvolupat per Microsoft Research, s'utilitza per crear classificacions i emparellaments de jugadors en la plataforma Xbox LIVE, i ha demostrat ser útil en el càlcul d'habilitats i classificacions en esports com el tennis, el golf o els escacs. TrueSkill és un algoritme d'inferència bayesiana, que utilitza tècniques de 'Message Passing' i 'Expectation Propagation for Message Approximation' en 'Factor Graphs' per construir un model capaç d'inferir les habilitats per cada equip o jugador. En aquest projecte s'exposen diferents implementacions del model TrueSkill, amb l'objectiu de calcular les habilitats dels jugadors o equips utilitzant les dades de la temporada 2019/2020. Després, aquestes habilitats s'utilitzen per realitzar la predicció de cada partit de la temporada 2020/2021. Finalment, la distancia i coeficient de correlació Kendall Tau s'utilitzen com a mesures d'avaluació, comparant els resultats obtinguts, tant en la inferència com en la predicció, amb les veritables classificacions.

1.3 Spanish version

Este proyecto introduce y testea el modelo de TrueSkill, con el objetivo de calcular las habilidades y de desarrollar un modelo predictivo tanto a nivel de jugadores como a nivel de equipos de la liga profesional de básquet NBA. TrueSkill, desarrollado por Microsoft Research, se utiliza para crear clasificaciones y emparejamiento de jugadores en la plataforma Xbox LIVE, y ha demostrado ser de utilidad en el calculo de habilidades y clasificaciones en deportes como el tenis, el golf, o el ajedrez. TrueSkill es un algoritmo de inferencia bayesiana, que utiliza técnicas de 'Message Passing' y 'Expectation Propagation for Message

Approximation' en 'Factor Graphs' para construir un modelo capaz de inferir las habilidades de cada equipo o jugador. En este proyecto se exponen diferentes implementaciones del modelo de TrueSkill, con el objetivo de calcular las habilidades de los jugadores o equipos usando datos de la temporada 2019/2020. Después, dichas habilidades se usan para realizar la predicción de cada partido de la temporada 2020/2021. Finalmente, la distancia y el coeficiente de correlación Kendall Tau se utilizan a modo de evaluación para comparar los resultados obtenido en la inferencia y predicción con las clasificaciones verdaderas.

Contents

2 Mathematical Background

In this section the TrueSkill method and the algorithms used to develop it are briefly described. TrueSkill is a Bayesian inference algorithm, which implements the Sum-Product Algorithm to perform message passing in the factor graph. The Expectation Propagation algorithm is introduced to do approximate inference. The inference problem consists in computing a posterior probability over a set of variables, in this case rankings, and it is in general intractable for exact computation. Finally, the basic notion of Kendall Tau distance is introduced, as it is the metric measure used to evaluate every implemented model, comparing both ground truth and predicted ranking by each model.

2.1 Bayesian Inference

As previously mentioned, TrueSkill [8] is a Bayesian inference algorithm. Bayesian inference is a statistical inference method constructed around Bayes' theorem. Bayes theorem relates the probability of an event based on factors related to such event, so it is used to update the posterior probability for a hypothesis or event after the fact that some evidence is observed.

Being A and B random events, the following formula summarizes Bayes' theorem:

$$P(A|B) = \frac{P(B|A)}{P(A)P(B)}$$

.

In Bayesian inference [13], $P(A)$ probability is updated according to Bayes' theorem. $P(A)$, also known as prior probability, represents the probability of the hypothesis before some evidence occurs, $P(A|B)$, also known as posterior probability, is the hypothesis probability after new evidence is observed, and $P(B|A)$, also known as likelihood, is the probability of the evidence once the hypothesis is observed.

2.2 Bayesian network

Bayesian networks are a type of directed acyclic probabilistic graphical model, which are useful to represent and understand the relationship between a set of variables and its conditional dependencies in a probabilistic model. As one could conclude from its name, Bayesian network models use Bayesian inference algorithm to compute probability distributions.

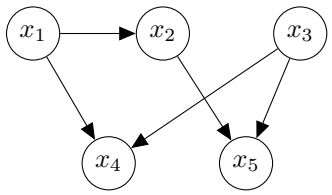

Figure 1: Bayesian network representing a joint probability distribution defined over five random variables $x_1, \ldots, x_5$, where each node represents a variable and every directed edge a conditional relation between nodes.

A graphical model such as the Bayesian network consists of nodes, representing variables, and directed edges, representing conditional

dependencies, and thus a probabilistic relationship, between variables. The direction of the edge represents the dependency between the variables involved. In this way, the node where the edge points to, also known as the child node, is said to be conditionally dependent on the node where the edge comes from, also known as the parent node. For example, as we see in Figure 1, there is an edge pointing out from x_1 to x_2, so the conditional distribution involving both variables is $P(x_2|x_1)$, and x_2 is conditionally dependant on x_1, as the probability distribution of x_2 event depends on the probability of the event x_1.

By using Bayesian networks, and because the local Markov property about a node conditionally independent of any other non-descendant nodes holds, we are able to simplify the joint distribution calculation. The joint distribution for a Bayesian network is described by,

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} P(x_i|parents(x_i))$$

.

Considering the Bayesian network represented in Figure 1, the joint distribution is computed as,

$$P(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_3)p(x_2|x_1)p(x_4|x_1, x_3)p(x_5|x_2, x_3)$$

.

Therefore, Bayesian networks can be used to represent efficiently very large joint probability distributions. In this case of the example, assuming

binary variables, one only needs to define a few factors involving at most three variables (2^3 values), whereas the full joint distribution is defined for 2^5 values.

2.3 Factor graph

Factor graphs [1] are also a type of probabilistic graphical model, which is mainly used to develop inference algorithms. In TrueSkill, factor graphs are applied as part of Bayesian inference algorithms. Factor graphs are an alternative representation of a factorized distribution that defines explicitly the factors as nodes in the graph. A factor graph is a bipartite graph useful to represent the factorization of a probability distribution function.

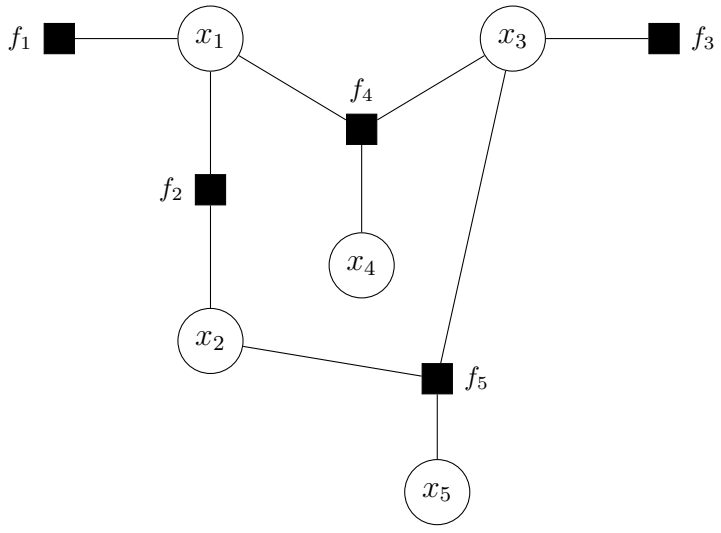

Figure 2: Factor graph representing a joint probability distribution, where each variable is represented by a circle node, and each factor by a black square node.

Given a function $f(x_1, ..., x_n) = \prod_i f_i(x_i)$, the factor graph contains a node for each factor f_i and each variable x_i in the formula. If the local factors $f_i(x_i)$ are not normalized, i.e., they do not sum up to one, the product does not define a probability distribution. Because of that one needs to multiply by a normalizing factor Z that sums over all possible configurations of the variables.

Therefore, every factor graph is compounded by two types of nodes, variable nodes represented by a circle and factor nodes represented by a square, and edges in the graph represent the dependency of factors on variables. Remember that a factor graph is bipartite, so every edge in a factor graph links different types of nodes, so that they always link a factor node with a variable node.

Figure 2 represents a factor graph structure. Such representation is equivalent to the Bayesian network represented in Figure 1. In this case, the joint probability distribution is computed as,

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{1}{Z} f_1(x_1) f_3(x_3) f_2(x_1, x_2) f_4(x_1, x_3, x_4) f_5(x_2, x_3, x_5)$$

2.4 Message Passing. Sum-Product Algorithm on factor graphs

Let's remind ourselves for a moment about our goal. With TrueSkill, what we aim to do is to compute the marginal prior probability of a hypothesis after some new evidence is provided. Once a factor graph is used

to represent the factorization of such a probability function, Message Passing [2], or Sum-Product Algorithm, enables us to compute marginals and conditionals in an efficient way, by passing messages on the factor graph from one node to another. Thus, TrueSkill uses the Sum-Product Algorithm to perform message passing in the factor graph.

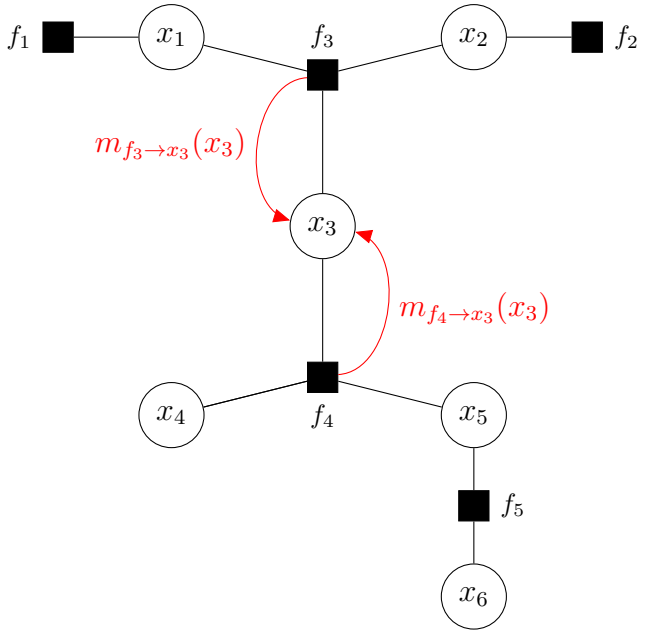

Figure 3: Factor graph where message passing is applied. Each variable x_i is represented with a variable node, and each factor f_i with a factor node. Messages $m_{f_3 \to x_3}$ from factor node f_3, and $m_{f_4 \to x_3}$ from factor node f_4 are shown to compute marginal probability of node x_3.

The key point of the Sum-Product Algorithm efficiency remains on how the information is kept locally, reducing marginals computational cost. The idea is to perform local computation, known as messages, on a

factor graph. Each message (factor) is associated with each edge in both directions. Initially, messages from leaf node factors are initialised to the factor, and messages from leaf variable nodes are set to one. Then, iteratively, information is propagated along the edges of the factor graph via message update equations, until convergence is achieved. When this happens, we then are able to easily compute the marginal probabilities at each node, by taking the product of the incoming messages to the node.

There are three types of possible update equations:
First of all, messages from variable node to factor node are computed by the product of every incoming message from the set of neighbors factor nodes of the sender variable, except the receiving one (the factor node to which the message is sent).

$$m_{x_k \to f_s} = \prod_{i \in F(x_k) \setminus f_s} m_{f_i \to x_k}(x_k)$$

Secondly, messages from factor node to variable node are computed as the product of every incoming message from the set of neighboring variable nodes of the sender factor, except the receiving one (the variable node to which the message is sent). This is multiplied by the corresponding factor, and marginalized over all the variable nodes linked to the factor node.

$$m_{f_s \to x_k} = \sum_{x_1} \sum_{x_2} ... \sum_{x_m} f_s(x_s) \prod_{i \neq j} m_{x_i \to f_s}(x_i)$$

Finally, marginals at a variable node are computed as the product of all incoming messages from its neighbour factors.

$$p(x_k) = \prod_{i \in F(x_k)} m_{f \to x_k}(x_k)$$

As we can see in Figure 3, the Sum-Product algorithm allows us to compute, for example, variable node x_4 marginal probability with a lower computational cost. To illustrate this computational reduction, let's first compute marginal probability in node x_4 without using the message passing algorithm. By following the factor graph formulation, such marginal would be computed as

$$p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \sum_{x_6} f_1(x_1) f_2(x_2) f_3(x_1, x_2, x_3) f_4(x_3, x_4, x_5) f_5(x_5, x_6)$$

Having N possible states for each variable, the computational cost of that operation would be N^6. Contrarily, by using the message passing we are able to reduce the computational cost of such operation to N^3. Sum-Product algorithm for marginal probability computation of node x_3 is:

$$\begin{aligned} p(x_3) &= \frac{1}{Z} m_{f_3 \to x_3}(x_3) \cdot m_{f_4 \to x_3}(x_3) \qquad (1) \\ &\propto \Big(\sum_{x_1} \sum_{x_2} f_1(x_1) f_2(x_2) f_3(x_1, x_2, x_3) \Big) \Big(\sum_{x_4} \sum_{x_5} \sum_{x_6} f_4(x_3, x_4, x_5) f_5(x_5, x_6) \Big) \\ &\propto \Big(\sum_{x_1} \sum_{x_2} f_1(x_1) f_2(x_2) f_3(x_1, x_2, x_3) \Big) \Big(\sum_{x_4} \sum_{x_5} f_4(x_3, x_4, x_5) \Big(\sum_{x_6} f_5(x_5, x_6) \Big) \Big), \end{aligned}$$

where $Z = \sum_{x_3} m_{f_3 \to x_3}(x_3) \cdot m_{f_4 \to x_3}(x_3)$.

2.5 Expectation Propagation

In some situations, messages in Sum-Product Algorithm can not be represented in a compact form, and we need to use some kind of approximation when performing message passing. Expectation Propagation plays a key role when implementing TrueSkill, as it infers the team, or players (depending on which implemented approach described later), skills and probabilities of winning a game. Furthermore, it also converts the intractable truncated game outcome to a tractable Gaussian outcome. When implementing TrueSkill we initialize priors as Gaussians, but to also get a Gaussian posterior distribution we need to use Expectation Propagation. To achieve a posterior approximation distribution which mimics the true posterior distribution, we need to define a Gaussian distribution using the same mean and variance as the true posterior non-Gaussian distribution has.

When implementing TrueSkill, we need to use approximate inference because we are building up a factor graph which is not a tree, and furthermore Expectation Propagation because variables are not discrete but continuous.

Assume we reach an intractable posterior probability distribution. The idea in Expectation Propagation [3] is to optimally select those factors which, when replaced by simpler approximating factors, would convert the

distribution to a tractable one. The approximate factor used to replace its true factor is the one which minimizes Kullback-Leibler divergence. Kullback-Leibler divergence measures the difference between two distributions, and therefore is used in the Expectation Propagation algorithm to determine how close the approximated distribution is to the true one. If all approximation factors are correctly set, then when replacing the approximation factor by its corresponding factor, no change in marginals calculation would be noticed.

The Expectation Propagation algorithm for approximate inference problems with continuous Gaussian variables is summarized by Minka [4] with the following representation.

Algorithm 1 Expectation Propagation for Approximate Inference Problems

1: Initialization performed as variance $v_i = \infty$, mean $m_i = 0$, and $s_i = 1$. Term approximation initialization as $t_i(w) = s_i \exp -\frac{1}{2v_i}(w^T x_i - m_i)^2$

2: Factors defined as $q(w) = \mathcal{N}(M_w, V_w)$, using $M_w = 0$ and $V = I$ as priors.

3: Until every mean m_w and variance v_w converge:

Remove approximate t_i from the posterior to get an old posterior and compute:

$$V_w^{/i} = V_w + \frac{(V_w x_i)(V_w x_i)^T}{v_i - x_i^T V_w x_i}$$

$$M_w^{/i} = M_w + (V_w^{/i} x_i) v_i^{-1} (x_i^T M_w - m_i)$$

Recompute M_w and V_w using $M_w^{/i}$ and $V_w^{/i}$ respectively.

$$z_i = \frac{(M_w^{/i})^T x_i}{\sqrt{x_i^T V_w^{/i} x_i + 1}}$$

$$\alpha_i = \frac{1}{\sqrt{x_i^T V_w^{/i} x_i + 1}} \frac{\mathcal{N}(z_i, 1)}{z_i}$$

$$M_w = M_w^{/i} + (V_w^{/i} \alpha_i x_i)$$

$$V_w = V_w^{/i} - (V_w^{/i} x_i) \left(\frac{\alpha_i x_i^T M_w}{x_{iT} V_w^{/i} x_i} \right) (V_w^{/i} x_i)^T$$

here $M_w^{/i}$ and $V_w^{/i}$ represent old posteriors mean and variance, computed as the product of messages into x_k except for those originating from term t_i.

Update t_i approximates by:

$$v_i = x_i^T V_w^{/i} x_i (\frac{1}{\alpha_i x_i^T M_w} - 1)$$

$$m_i = x_i^T M_w^{/i} + (v_i + x_i^T V_w^{/i} x_i) \alpha_i$$

$$s_i = \frac{z_i \sqrt{1 + v_i^{-1} x_{iT} V_w^{/i} x_i}}{\exp(-\frac{1}{2} \frac{x_i^T V_w^{/i} x_i}{x_i^T M_w} \alpha_i)}$$

Then, once posterior probability is achieved, conversion is computed as

$$p(D) \neq |V_w|^{1/2} \exp(\frac{B}{2}) \prod_{i=1}^{n} s_i$$

where $B = M_w^T V_w^{-1} - \sum_i \frac{m_i^2}{v_i}$.

3 TrueSkill Model

TrueSkill is the main method used in this paper. TrueSkill [7] is a Bayesian inference algorithm which provides a rating system among game players. Developed by Microsoft Research, it has been used for ranking and player matchmaking on Xbox LIVE. Additionally, it has been proven to be useful to create rankings in other sports such as tennis, golf or chess. The aim of this research is to use the TrueSkill model to create rankings in basketball, but more concretely in the NBA basketball league.

To compute the posterior win probability of a team, TrueSkill model uses factor graph representation and approximate message passing. The following explanation summaries the main points described in TrueSkill paper [8], applied to NBA basketball approach.

For each team t in the competition, there is a set of players n_t. Each player has its own different skill, represented as a Gaussian distribution $\mathcal{N}(\mu, \sigma)$. Each player in every game is considered to have a performance p_{n_t}, centered at Gaussian skill s_i and with a β variance. Then, each team performance p_t is computed as the aggregation of every player n_t. To represent such a relation between players and teams, and to efficiently compute its posterior probabilities, factor graphs are used.

Figure 4 represents a TrueSkill factor graph with approximate message passing model. In this example, only two teams, with three players

each, are considered because of simplification. In the complete factor graph for NBA TrueSkill model, there are more than 500 players grouped into 30 teams. As we can see in the figure, prior players' skills are represented as marginal factor nodes, which are related to players skills and performance, represented as variable nodes. We can also observe that players performances are then combined to obtain teams' performance. Finally, teams are combined between them to see teams' performance difference. Bent edges represent approximate message passing algorithm.

Particularly, in the NBA model, only two teams are involved in every match. Every time a match takes place, new evidence is provided to the factor graph, as the winner of such game is provided. To update players marginal skills because of this new evidence, approximate message passing technique is used. Sum-product algorithm is used to exploit the sparse connection structure of the graph, enabling efficient inference on marginal variables, which are prior skills of every player involved in such game.

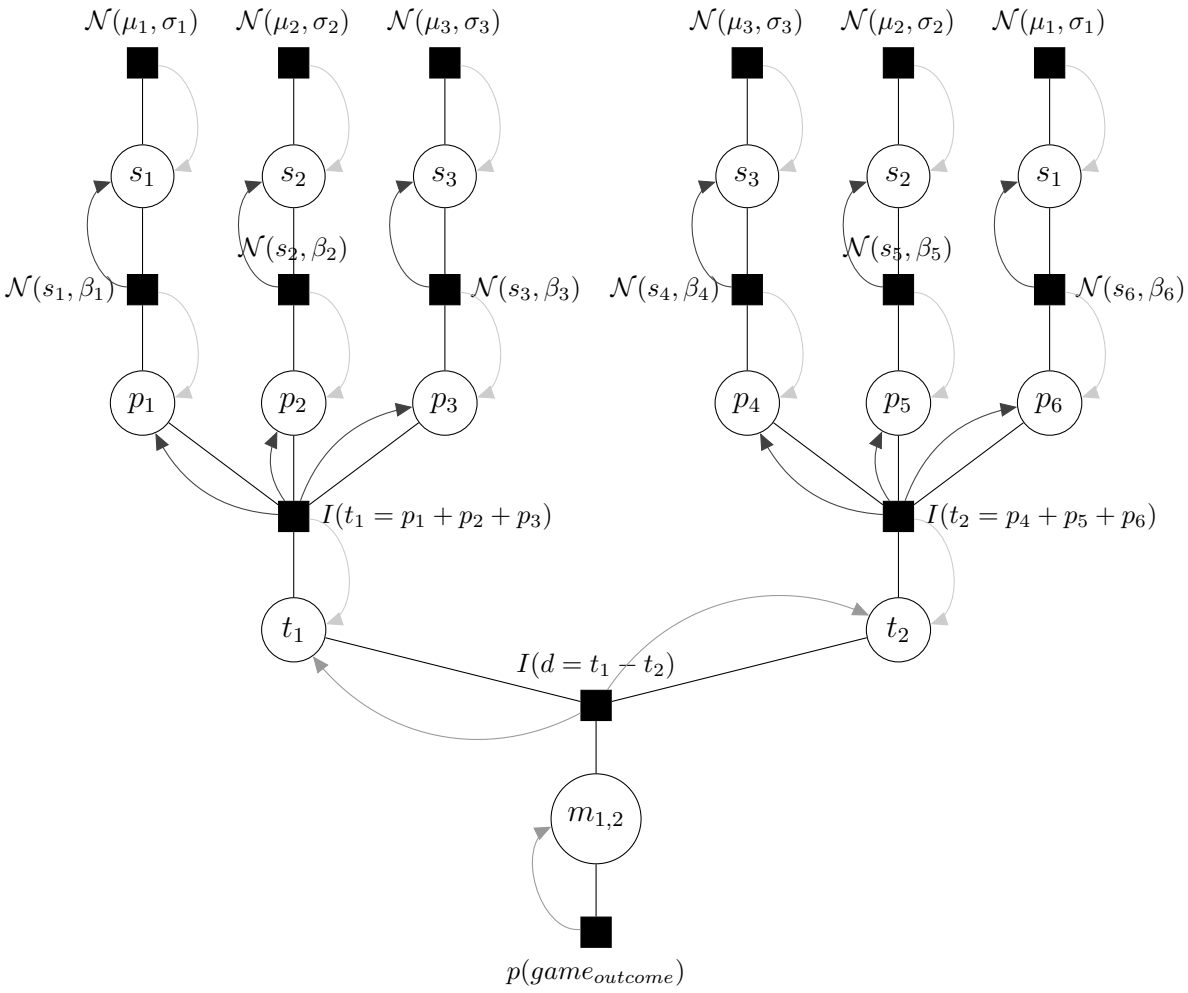

Figure 4: Representation of a TrueSkill factor graph with approximate message passing model. In this example, only two teams formed by three players each is considered.

4 Datasets

In this section, datasets used during this project [1] are explained. Datasets were extracted from Basketball Reference [5] and official NBA [6] websites.

4.1 Prior initialization datasets

The first dataset preparation is to do prior initialization. Depending on the approach, I would need either team or players individual statistics. Firstly, I used the official NBA website to obtain data for each teams' prior initialization. I was able to create a dataset including a set of average statistics of the 2018/2019 regular season for every team in the NBA. The goal is to perform inference on the 2019/2020 regular season games, so I needed to use information which was available before the start of the season.

Secondly, to recap data for players' prior initialization approach, I used the Basketball Reference website. Again for the 2018/2019 season, the dataset includes the average of a set of individual statistics for each player in any team in the league.

4.2 Games datasets

To do inference and prediction, I needed a dataset containing every game of the NBA regular seasons 2019/2020 and 2020/2021 respectively,

[1] Link to GitHub repository `https://github.com/apujol8/TFG.git`

	Visitor/Neutral	Home/Neutral	AwayWinner	HomeWinner
0	NOP	TOR	1	0
1	LAL	LAC	1	0
2	CHI	CHO	1	0
3	DET	IND	0	1
4	CLE	ORL	1	0
...	...	...	...	...
1054	NOP	ORL	1	0
1055	DEN	TOR	1	0
1056	MIA	IND	1	0
1057	OKC	LAC	1	0
1058	PHI	HOU	0	1

(a) 2019/2020 regular season dataset

	Visitor/Neutral	Home/Neutral	AwayWinner	HomeWinner
0	GSW	BRK	1	0
1	LAC	LAL	0	1
2	CHO	CLE	1	0
3	NYK	IND	1	0
4	MIA	ORL	1	0
...	...	...	...	...
1075	DAL	MIN	1	0
1076	LAL	NOP	0	1
1077	LAC	OKC	1	0
1078	DEN	POR	1	0
1079	UTA	SAC	0	1

(b) 2020/2021 regular season dataset

Figure 5: Games datasets display. There is a row for each game in each season.

and the outcome of each of those games. In the Basketball Reference website, I was able to find monthly games databases, for each month of the season, containing information about each game (home team, visitor team, the score of both teams, and other useless attributes). Therefore, I then was able to concatenate each of those files in a single database which contains a row for every single game for each regular season. The 2019/2020 regular season database includes 1059 rows, whereas the 2020/2021 regular season database includes 1080 rows, one for each game.

Figure 5 represents both created databases. As observable, each final database is composed of 4 attributes. The first two categories contain the visitor team code for each game, named ‘Visitor/Neutral’, and the home team code for each game, named ‘Home/Neutral’. Finally, the dataset includes two binary complementary categories, expressing either if it was

the home team or the visitor team that won each game. Columns are named $AwayWinner$ and $HomeWinner$. For each row in each database, if it was the home team that won the game, $AwayWinner$ is set so 1, whereas $HomeWinner$ is set to 0. One could think that this is the opposite way to arrange values, so that the winner should be set as a 1. The reason why I decided to do the opposite is because when using the TrueSkill algorithm to do inference, the considered winner of each game is the one with lower rank.

4.3 Actual ranking after the season datasets

Finally, I used the official NBA website to create actual ranking after each season, or ground truth datasets. I organized data to create a dataset for each 2019/2020 and 2020/2021 team rankings at the end of the regular season. The 2019/2020 ranking is used to compare against the ranking achieved after inference, whereas the 2020/2021 ranking is used to compare against the ranking achieved after prediction model. In addition, I also created a ranking of the best ten players during the 2019/2020 regular season, to compare it with the players ranking after inference.

5 Methods

In this section every stage to construct the TrueSkill model is introduced. Such stages are prior initialization, inference model, and prediction model. Moreover, results comparison techniques are explained for both teams and players skill rating units approaches.

5.1 Prior initialization

Regarding prior initialization in TrueSkill, $Rating(\mu, \sigma)$ is the method used to model the skill of a player. Players skills are represented as Gaussian distributions $\mathcal{N}(\mu, \sigma)$, where mu μ describes the prior average skill of the player, and sigma σ represents the uncertainty level of the average skill. Therefore, each player or team, depending on the approach, needs to be initialized with an initial μ and σ scores to define its skill rating function. Better teams or players will have big μ but low σ values, in comparison with worse teams or players.

In collective sports such as basketball, it is difficult to estimate which are important attributes for a player or team, as there are too many factors which, in some way, influence the game outcome. Thus, despite there being no definite truth about it, after some research and my basic understanding of how certain factors influence the game, I conclude that the following initialization is sufficient for a basic prior knowledge initialization.

To define priors, or initial rating units, we will divide our hypothesis into two different groups depending on the approach. Such groups are rating initialization for players and rating initialization for teams. Thus, further analysis on both team and individual approaches on rating initialization is performed to see which is the better prior initialization in each case, for both average skill μ and level of uncertainty σ. Each proposed hypothesis here is later discussed in the section Results.

5.1.1 Team rating initialization

On the one hand, average skill or μ initialization for the first described approach above was performed using collective team statistics of the overall 2019/2020 regular season. By trial and error and again my basic understanding of the game, I realised that two plausible average skill μ initialization approaches are either to use each team's average points per game, or a linear combination of averaged points, rebounds and assists.

On the other hand, it was more difficult to search for a suitable initialization for uncertainty σ. In the TrueSkill documentation, it is recommended to initialize σ with a third of the value which μ has. This recommendation is too conservative for my purposes, as it results in unrealistic differences between the uncertainties of teams that have very different values of μ. From my point of view, that is not a suitable enough initialization. Let me explain myself. If I would have initialized σ as recommended, then players or teams with higher μ, so with higher average skill values, would also have higher σ values, and thus would mean that there would be more

uncertainty with those top league players or teams than with other players or teams with lower initialized average skill.

For example, this would suppose that there would be more uncertainty about the four times NBA champion, 17 times All-Star and 4 times regular season MVP Lebron James' performance, who has been playing in the league for the last 18 seasons, than almost any other player in the league, just because his initial average skill was higher. The same applies to the team's initial average skill approach.

However, I finally decided to use the TrueSkill recommendation initialization for σ level of uncertainty as in the Results section it has been proven to work surprisingly well, as well as an average players' age inverse relation. In my opinion, experience is one of the most important factors when we are evaluating the confidence level of a team, so that the level of uncertainty of a team is inversely related to its players' average age.

5.1.2 Players rating initialization

Again, let's start discussing average skill initialization or μ. For individual player average skill initialization, both same approaches used for teams average skill initialization are considered, but another statistic seemed to better capture players average skill. These statistics were PIE, or Player Impact Estimate on game, and net rating. PIE is the combination of many individual statistics such as points, assists, rebounds, blocks, turnovers, etc, over the whole same statistics in the overall game, so it is a metric

able to capture a player's overall contribution to the game. Net rating is the average per game of the difference between scored and conceded points during the minutes in which the player is playing. Despite this, both PIE and Net Rating were discarded as possible initialization approaches, as both could include negative or very small positive μ and therefore σ initialization.

Secondly, when considering level or uncertainty or σ initialization, I considered again the same hypothesis as in the previous case, which was using an inverse relationship of the player's age to determine it, but also the TrueSkill recommended value with a slight modification. TrueSkill recommendation leaded into very small sigma values, so instead of using a third of μ for level of uncertainty initialization, I decided to use $\frac{\mu}{2}$, to obtain higher σ values. Recall that σ represents the level of uncertainty, and having higher values gives wider margin for learning in inference process.

Other initialization such as amount of games played, players role in the team, or leadership capability were considered. But as the first is related to players' age, and the other two are subjective and thus difficult to represent in an empirical way, they were finally discarded for further analysis.

5.2 Inference model

Once prior initialization is performed, what is next is to construct a model where each player or team, depending on the approach, is trained using

the labeled games in the 2019/2020 regular season.

Inference is performed using the TrueSkill $rate()$ method. What this method does is recalculate rating units of both teams, or teams' players, involved in a game, knowing in advance whoever the winner is. Thus, we need to know as parameters both teams or teams players involved in a game, and the outcome of such a game to be able to update teams or players skills. Therefore, here again different implementations are used depending on the approach.

5.2.1 Teams rating inference

In this approach, either teams are initialized as single rating units, or players are initialized as single rating units but arranged in a single team skill rating unit before inference process. Therefore, when using the method $rate()$ to do inference for each game in the 2019/2020 NBA regular season, both teams rating skill unit and the outcome, or ranks, for each of them are provided as parameters. For each match, level of uncertainty σ is reduced for both teams, and average skills μ are recomputed based on the outcome of such a game. Skill value of the winning team increases, whereas there is a decrease in the loser's team skill value.

In the end, a ranking of teams in the NBA, ordered from best to worst skill is obtained. Remember that every team is represented as a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, so that teams in such a ranking are ordered by $\mu - 2\sigma$.

5.2.2 Players rating inference

In this second approach, players are initialized as single rating units. Therefore, when using the method $rate()$ with inference purpose, for each game in the 2019/2020 NBA regular season, players rating skill units list for both teams involved in addition to the outcome, or ranks, are passed to the method as parameters. Here, the method works by reducing the uncertainty σ of every player involved in the game, and average skills μ are updated. Players in the winning team μ gain skill, whereas players in the losing team lose it.

Regarding ranking construction, players are ranked in the same way, following $\mu - 2\sigma$ value in decreasing order.

5.3 Prediction model

After inference process, where every single game in the 2019/2020 season is processed, players or teams learned skills are used to make predictions of future games. In this case, skills were used to predict the outcome for every game in the 2020/2021 regular season. Thus, for each game in the 2020/2021 season, probability of win for both involved teams is computed for each game, and afterwards the outcome of such game is produced using a random uniform number generator. To compute each teams' win probability involved in a game, the cumulative distribution function over the difference between the players skills aggregation of both teams is used.

The prediction model is implemented in the same way for all possible described approaches. To compute win probability for each game, the method receives as parameters both involved teams' skills rating units. This means that, if needed, players rating units need to be arranged in a single team skill rating unit before prediction model. This only occurs when inference is performed using players skill rating unit.

The prediction model is conducted over 1000 different season simulations, where at the end of each a dictionary including each teams' amount of wins is obtained.

5.4 Kendall Tau distance

To evaluate the prediction model, where a ranking of NBA teams is created according to every game prediction, Kendall Tau distance and correlation metrics are used. Kendall Tau distance [9] is a metric which counts the amount of pairwise disagreements between two ranking lists. Therefore, Kendall Tau distance is applied to see how similar the predicted ranking is in relation to the ground truth ranking for the 2020/2021 NBA regular season.

Kendall Tau distance between two rankings r_1 and r_2 is computed as:

$$K(r_1, r_2) = \frac{1}{\frac{n(n-1)}{2}} \Big| (i,j) : i < j, \Big(\big(r_1(i) < r_1(j)\big) \wedge \big(r_2(j) < r_2(i)\big) \Big) \vee \\ \vee \Big(\big(r_1(j) < r_1(i)\big) \wedge \big(r_2(i) < r_2(j)\big) \Big) \Big| \quad (2)$$

being n the number of elements in the rankings.

The Kendall Tau Distance counts the number of times the values in ranking r_1 are in the opposite order of the values in ranking r_2. Thus, if rankings r_1 and r_2 are identical, $K(r_1, r_2)$ will equal 0, whereas it will equal 1 if r_1 is the reversed of r_2. Therefore, more similar rankings would have a smaller Kendall Tau distance value rather than less similar rankings.

In addition, the Kendall rank correlation [10], or τ, coefficient was also used to compare both predicted and ground truth rankings. The Kendall τ coefficient checks for every position in the ranking if both contain the same element, and thus are concordant pairs, or if contain a different element and are discordant pairs, is computed as:

$$\tau = \frac{number\ of\ concordant\ pairs - number\ of\ discordant\ pairs}{\frac{n(n-1)}{2}}$$

Two rankings r_1 and r_2 will have a moderate correlation if its Kendall τ correlation value is above $0'2$, whereas if the value exceeds $0'3$ the correlation is considered strong. Lower τ values suggest weak correlation.

6 Implementations

This section describes three different TrueSkill models designed to create an NBA team ranking for the 2020/2021 regular season [2]. The following table summarizes each approach by describing how priors initialization, and inference and prediction models are performed by one of the explained methods in the previous section.

	Approach 1	**Approach 2**	**Approach 3**
Prior initialization	Teams rating initialization	Players rating initialization	Players rating initialization
Inference model	Teams rating inference	Teams rating inference	Players rating inference
Predictive model	Teams rating prediction	Teams rating prediction	Teams rating prediction

6.1 Approach 1. Team as a rating unit

The first approach, and the most general one, consists of considering every team in the league as a single rating unit. In this way, there are only 30 players or rating skills in the model, one each for every team in the league. to perform prior initialization, the above described teams rating initialization is used. Therefore, each team is initialized using some collective team statistic. To implement inference and predictive models, also team rating approach is used. Teams skills are estimated performing

[2]Link to GitHub repository `https://github.com/apujol8/TFG.git`

inference using the NBA 2019/2020 regular season schedule, and then such learned skills are used to predict the outcome of every game in the 2020/2021 regular season.

6.2 Approach 2. Players as rating units, inference on teams

The second approach is based on the idea that teams' skills are directly related to their best players' individual skills. Therefore better players stats aggregation would better capture a teams skills in comparison with collective team statistics used in the first approach. Considering every team as a set of players, every team skill is constructed from its three better players' individual stats aggregation. Thus, prior initialization is performed using players rating initialization approach, so that we have a skill rating for each player in the league, which are then combined and grouped into teams to achieve a skill for each team before inference and prediction. The teams skills are estimated performing inference using the NBA 2019/2020 regular season schedule, and then such learned skills are used to predict the outcome of every game in the 2020/2021 regular season. An important issue to notice is that teams are usually formed from 12 up to 16 players. In my analysis, I decided to reduce the number of players in each team to 3. I decided to do so because, from my point of view, big differences in teams' performance fall on its best players' differences rather than role and secondary players.

6.3 Approach 3. Players as a rating unit

Finally, the last approach is also based on the idea that teams' skills and players' individual skills are directly related to each other. So again, prior initialization is performed as players rating units, so that there is a rating skill for each player in the league. But the difference with the previous approach is that here players' individual skills are not combined into teams before performing the inference process. Instead, inference is performed with players rating inference approach. Players are the ones trained here, not teams. N to N matches, being N the number of players in a team are ruled. So after the inference, what is obtained is not a ranking of teams but a ranking of players in the league. Therefore, this approach is also useful to consider season MVP prediction. Then, to perform predicted team rankings for the 2020/2021 regular season, players are rearranged in teams before the prediction model. In this way, the model is able to capture player transfers from one team to another from one season to another.

7 Results and discussion

In this section, each of the previously introduced implementations is tested and briefly discussed.

7.1 Approach 1

7.1.1 Prior initialization

To evaluate team rating units prior initialization, 2018/2019 regular season ground truth ranking is compared with the ranking after each initialization choice using normalized Kendall Tau distance and τ correlation coefficient metric measures.

The following tables summarize each μ and σ possible initialization combination for both normalized Kendall Tau distance and τ coefficient correlation. Average points, a combination of average points, assists and rebounds, and TrueSkill predefined value (which is 25), are considered as possible μ values. For σ values, two considered approaches are the TrueSkill recommended value ($\frac{\mu}{3}$), and $\frac{\mu}{2} - AvAge$ representing an inverse average age relation.

Table 1 corresponds to normalized Kendall Tau distance measure. As we can see, best possible initialization is when using a combination between average points, rebounds and assists for μ, and TrueSkill recommended value for σ.

NKT distance	**mu** μ			
Sigma σ		Points	Pts+Ast+Rbd	Recommended
	Average age	0.34	0.35	0.66
	Recommended	0.29	**0.27**	0.44

Table 1: Normalized Kendall Tau (NKT) distance for every possible μ and σ combination.

τ **correlation**	**mu** μ			
Sigma σ		Points	Pts+Ast+Rbd	Recommended
	Average age	10.35%	**13.56%**	5.75%
	Recommended	2.99%	6.21%	1.15%

Table 2: τ coefficient correlation for every possible μ and σ combination.

Table 2 corresponds to Kendall τ coefficient correlation measure. Best possible initialization is again when using combination of average points, rebounds and assists for μ, and inverse average age relation for σ.

Contrary to σ initialization, both metric evaluation measures conduct to the same μ initialization, which is using a combination of the average points, assists and rebounds. Despite both metric measures leading to different σ initialization approaches, I finally decided to follow Kendall Tau distance´s best result, and thus initialize σ using TrueSkill recommended value. The main reasons to do so are that in this way we achieve higher sigma values before inference and thus more capability to learn skills, whereas otherwise, if other initialization options would have been chosen,

there would be teams with very low uncertainty values.

7.1.2 Inference model

Inference model is only tested using average points μ and TrueSkill predefined σ values, as has been proven to be the best possible prior initialization. Figure 6 represents sigma σ evolution through the inference, or learning, process. We are able to see that during inference, every team gets its σ, or performance uncertainty, reduced.

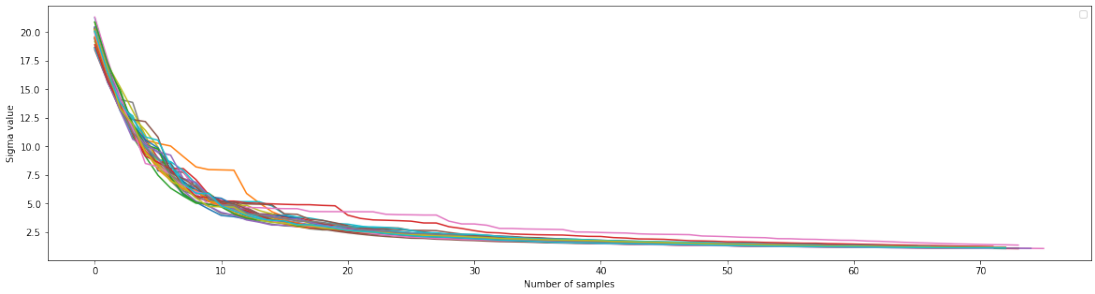

Figure 6: Each team's sigma σ evolution over inference process. Reduction is uncertainty: the more data is considered for inference, the more certain (smaller values of σ) is the model.

Inference is performed using games in 2019/2020 regular season. After inference process, 2019/2020 regular season ground truth ranking is compared with after inference team ranking, using both Kendall Tau distance and τ coefficient correlation metrics. Obtained results are 0.07 for normalized Kendall Tau distance, and 25.98% for τ correlation coefficient.

7.1.3 Prediction model

The prediction model consists of predicting the outcome of every game during 2020/2021 regular season. To evaluate performance and accuracy of such model, the normalized Kendall Tau distance, the τ coefficient correlation, and the ratio of mismatch between the predicted outcome and the actual on games metrics are used. Obtained ranking is compared with ground truth 2020/2021 regular season ranking. Prediction process over 2020/2021 season was computed for 1000 different season simulations, to obtain more solid results.

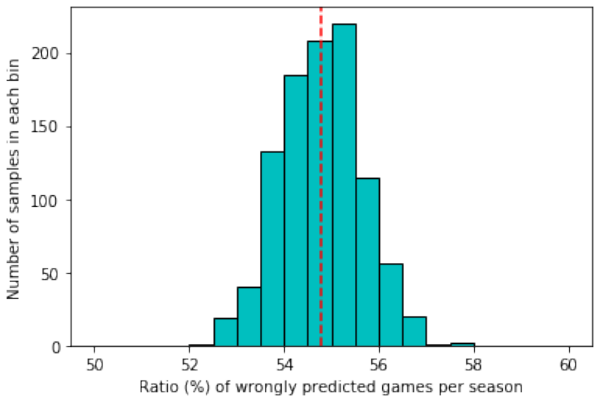

Figure 7: Histogram representing the ratio of wrongly predicted games per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 54.79.

Figures 7, 8, and 9 display results for above mentioned metric measures. Despite getting more than 50% wrongly predicted games per season, we observe a small Kendall Tau distance but a high τ correlation coefficient,

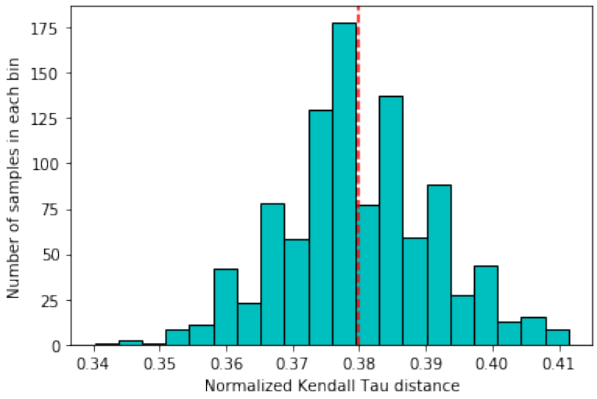

Figure 8: Histogram of predicted normalized Kendall Tau distance per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 0.38.

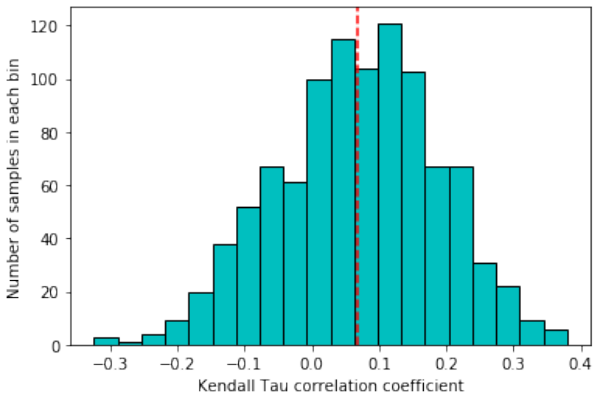

Figure 9: Histogram of predicted τ correlation coefficient per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 7.26%.

so that both measures conclude that predicted ranking and ground truth ranking for the 2020/2021 regular season are close to one another. For a better display of results, Figure 10 shows the average of predicted wins per season over 1000 different season simulations against the ground truth number of games won, for every team in the NBA league. The average difference amount of games won between predicted and real rankings is 15.65. We can thus conclude ...

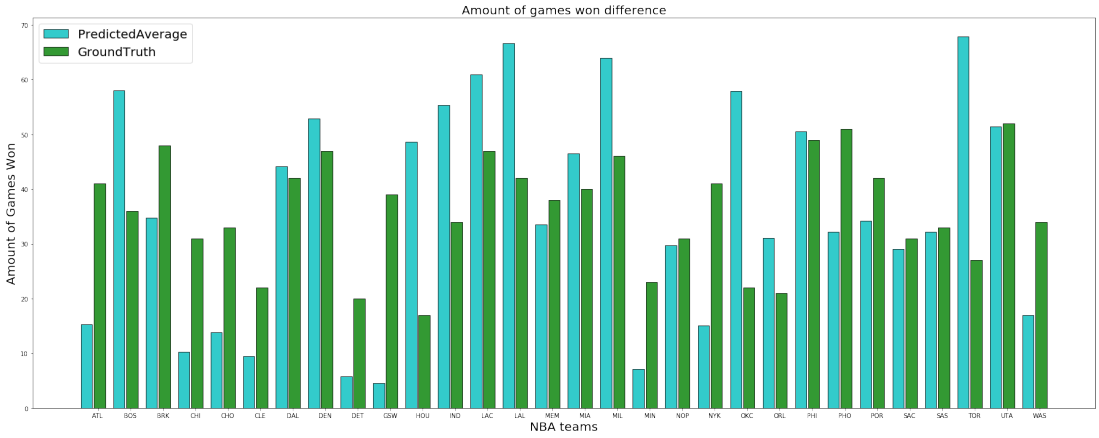

Figure 10: Histogram of predicted average wins per season, in blue, against the ground truth wins per team, in green. The average difference between both predicted and ground truth amount of win games for 2020/2021 regular season is 15.65.

As observed, obtained results on outcome game predictions are much better than expected, when comparing them to TrueSkill paper results on online games, but are slightly worse than results obtained in the paper [11], where other sports such football, tennis or golf are tested. Despite this, to

search for better prediction results, other approaches which apparently seem to better capture teams skills are tested.

7.2 Approach 2

7.2.1 Prior initialization

Prior initialization is evaluated using the same procedure and metric measures explained in Approach 1. The following tables summarizes each μ and σ possible initialization combination for both normalized Kendall Tau distance and τ coefficient correlation. Also in this approach, average points, a combination of average points, assists and rebounds, and TrueSkill predefined value (which is 25), are considered as possible μ values. For σ values, two considered approaches are the TrueSkill recommended one ($\frac{\mu}{3}$), and $\frac{\mu}{2} - Age$, representing an inverse age relation.

NKT distance	**mu** μ			
Sigma σ		Points	Pts+Ast+Rbd	Recommended
	Average age	0.53	0.56	0.65
	Recommended	0.39	**0.36**	0.44

Table 3: Normalized Kendall Tau (NKT) distance for every possible μ and σ combination.

For both Table 3, corresponding to normalized Kendall Tau distance, and Table 4, corresponding to to Kendall τ coefficient correlation measure, the best possible initialization is when using combination of average points,

τ **correlation**	**mu** μ			
Sigma σ		Points	Pts+Ast+Rbd	Recommended
	Average age	-21.84%	-15.40%	-14.94%
	Recommended	0.69%	**1.61%**	1.15%

Table 4: τ coefficient correlation for every possible μ and σ combination.

rebounds and assists for μ, and TrueSkill recommended value for σ.

7.2.2 Inference model

The inference model is only tested using only prior initialization which has been proven to be the best possible prior initialization. Figure 11 represents σ evolution through the inference or learning process. We are able to see that during inference, every team gets its σ, or performance uncertainty, reduced.

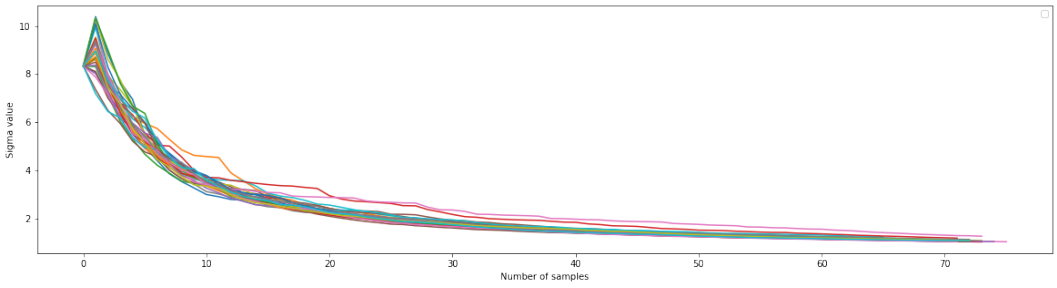

Figure 11: Each team's sigma σ evolution over inference process. Reduction is uncertainty: the more data is considered for inference, the more certain (smaller values of σ) is the model.

After the inference process, the 2019/2020 regular season ground truth ranking is compared with the after-inference team ranking, using both Kendall Tau distance and τ coefficient correlation metrics. Obtained results are 0.06 for normalized Kendall Tau distance, and 25.06% for τ correlation coefficient.

When comparing both approaches' results after the inference process, we observe almost no difference.

7.2.3 Prediction model

Here again, to evaluate the performance and accuracy of the prediction model, the same procedure and metric measures are used.

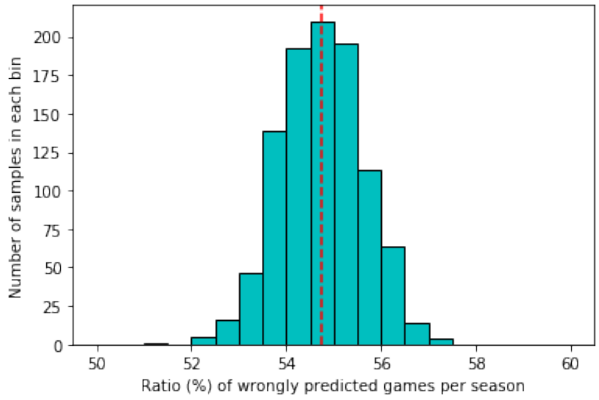

Figure 12: Histogram representing the ratio of wrongly predicted games per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 54.73.

Figures 12, 13 and 14 display results for above mentioned metric measures, and Figure 15 shows the average of predicted wins per season

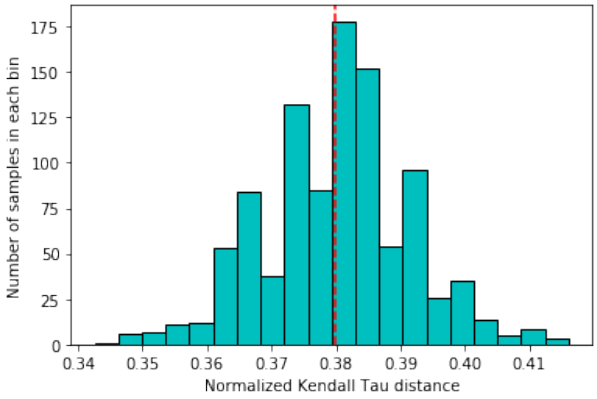

Figure 13: Histogram of predicted normalized Kendall Tau distance per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 0.38.

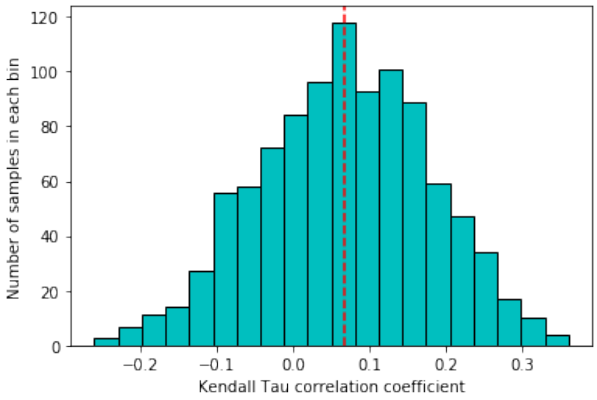

Figure 14: Histogram of predicted τ correlation coefficient per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 6.37%.

over 1000 different season simulations against the ground truth number of games won, for every team in the NBA league.

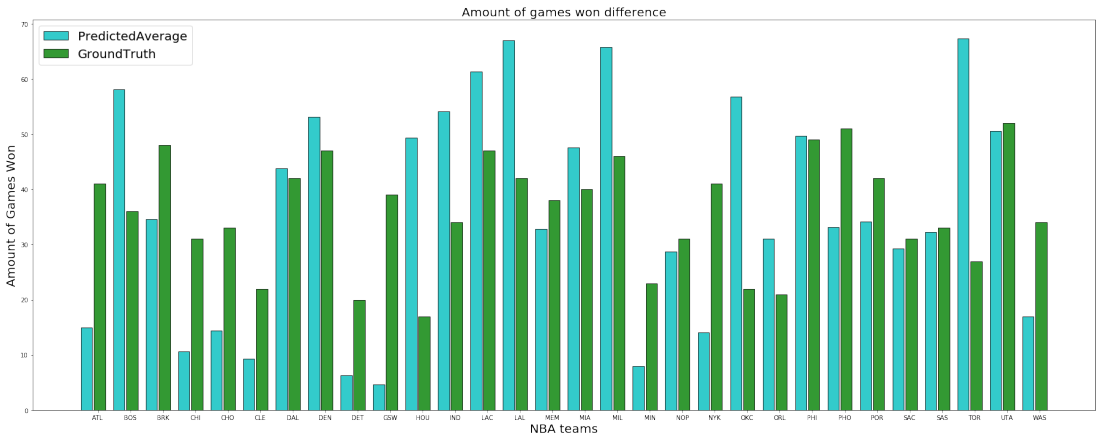

Figure 15: Histogram of predicted average wins per season, in blue, against the ground truth wins per team, in green. The average difference between both predicted and ground truth amount of win games for 2020/2021 regular season is 15.68.

In this second approach, results are slightly better for games misprediction rate and τ correlation coefficient. However, comparing both evaluated approaches there's almost no difference. The only difference between the first and second implemented approaches is how teams' prior skills is performed. Therefore, this gives an insight of how important prior initialization is for the model.

7.3 Approach 3

7.3.1 Prior initialization

In this case the inference model is performed over players. In TrueSkill, every player involved in a same team respond in the same way to the learning model (either if its a win or lose case). If every player in a team is initialized at a same skill, after the inference process they will all also have the same, but different from before inference, skill. Therefore, in this approach it is omitted to consider every initialization approach which would lead to same initial skills for players in the same team. To be able to compare rankings, players are grouped into teams, and then obtained ranking is compared with the ground truth ranking for 2018/2019 regular season.

However, in this approach, recommended σ initialization is changed to $\frac{\mu}{2}$, to have obtain greater σ values. Despite this, results are the same as in the previous approach, and thus the best possible initialization is when using combination of average points, rebounds and assists for μ, and $\frac{\mu}{2}$ for σ. All considered options for both μ and σ initialization are shown in Table 5, for normalized Kendall Tau distance measure, and Table 6, for τ correlation coefficient metric.

Moreover, particular players rating unit initialization evaluation was conducted. In this case, comparison was more difficult to be produced, as there is no ground truth database ranking players from best to worse.

NKT distance	**mu** μ			
Sigma σ		Points	Pts+Ast+Rbd	Recommended
	Average age	0.54	0.60	0.67
	$\frac{\mu}{2}$	0.60	**0.50**	-

Table 5: Normalized Kendall Tau (NKT) distance for every possible μ and σ combination.

τ **correlation**	**mu** μ			
Sigma σ		Points	Pts+Ast+Rbd	Recommended
	Average age	-5.75%	-5.75%	5.75%
	$\frac{\mu}{2}$	10.81%	**11.26%**	-

Table 6: τ coefficient correlation for every possible μ and σ combination.

Therefore, there is no method to compare players initialization apart from using MVP players list as the ground truth database. The drawback of this method is that such list contains only 10 best players for every season.

Comparing players initialization list with ground truth 2018/2019 MVP list, obtained results were -15.55% for τ correlation coefficient and 0.28 for normalized Kendall Tau distance. Obtaining a negative τ correlation coefficient can be attributed to the fact that, as seen in Figure 16, only 6 out of 10 players from ground truth ranking appear on the initialized one, and thus the number of concordant and discordant pairs can not be correctly computed. However, prior initialization seems to be accurate, as both major candidates to 2018/2019 regular season MVP, `Giannis`

Player	Rank
Giannis Antetokounmpo	49.0
James Harden	31.0
Paul George	61.0
Nikola Jokic	22.0
Stephen Curry	28.0
Damian Lillard	73.0
Joel Embiid	67.0
Kevin Durant	29.0
Kawhi Leonard	82.0
Russell Westbrook	62.0

(a) Ground truth 2018/2019 MVP ranking.

	Pos
James Harden	31
Giannis Antetokounmpo	49
Joel Embiid	67
LeBron James	40
Russell Westbrook	62
Anthony Davis	55
Paul George	61
Karl-Anthony Towns	52
Kevin Durant	29
Stephen Curry	28

(b) After initialization top 10 ranking.

Figure 16: Players initialization ranking evaluation.

`Antetokouonmpo` and `James Harden`, appear in the first two positions of the prior initialization ranking.

7.3.2 Inference model

The inference process is also considered using players rating units. Players are trained, not teams. To evaluate inference process, players are rearranged into teams so we can compare results with the ground truth 2019/2020 team ranking. Metric measures obtained results are 0.18 for normalized Kendall Tau distance, and 32.87% for τ correlation coefficient.

A problem that occurs from one season to another is new players appearance. Prior initialization is performed using 2018/2019 season average data, and inference is performed over 2019/2020 season.

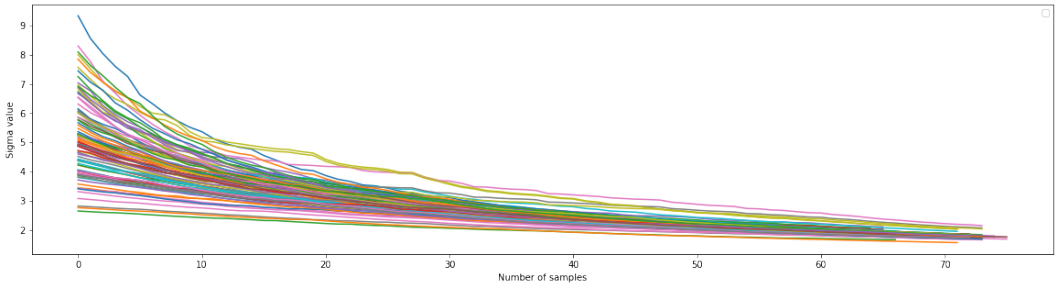

Figure 17: Each player's sigma σ evolution over inference process. Reduction is uncertainty: the more data is considered for inference, the more certain (smaller values of σ) is the model.

Therefore, in 2019/2020 season there may be, and there are, players who did not play during the previous season, and thus there is no prior knowledge of them. For such players, prior initialization is performed using 2019/2020 average data. Despite this, it is possible that some inconsistency arises in the model, as some players' initialization is performed using data which should only be available after inference. This is the only way to perform prior initialization for such rookie players.

Moreover, the inference model is also evaluated for individual rating units. Figure 17 represents players' sigma σ evolution through the inference, or learning, process. We are able to see that, as expected, during inference every players gets its σ, or performance uncertainty, reduced. Players ranking after inference is compared with ground truth 2019/2020 MVP ranking, and obtained metric results are 0.78 for normalized Kendall Tau distance, and -2.23% for τ correlation coefficient. In Figure 18, both predicted and ground truth 2019/2020 MVP rankings are shown one

Player	Rank
Giannis Antetokounmpo	97.0
LeBron James	79.0
James Harden	61.0
Luka Doncic	37.0
Kawhi Leonard	163.0
Anthony Davis	109.0
Chris Paul	64.0
Damian Lillard	145.0
Nikola Jokic	43.0
Pascal Siakam	164.0

(a) Ground truth 2019/2020 MVP ranking.

	Pos
Kemba Walker	25
Giannis Antetokounmpo	97
Nikola Jokic	43
Pascal Siakam	164
Kyle Lowry	166
Paul George	121
Joel Embiid	133
James Harden	61
Danilo Gallinari	74
LeBron James	79

(b) After inference top 10 ranking.

Figure 18: Players initialization ranking evaluation.

against another. As we see, only 4 out of 10 players appear in both rankings. Although `Giannis Antetokounmpo`, 2019/2020 regular season MVP winner, appears on the second position, in this case the ranking after inference differs more from the ground truth one.

7.3.3 Prediction model

Prediction is performed over team rating units. Beforehand, players are rearranged into teams. In this way, the model is able to capture players transfer movements from one team to another between 2019/2020 and 2020/2021 regular season. To evaluate performance and accuracy of the prediction model, again both the same procedure and metric measures are used.

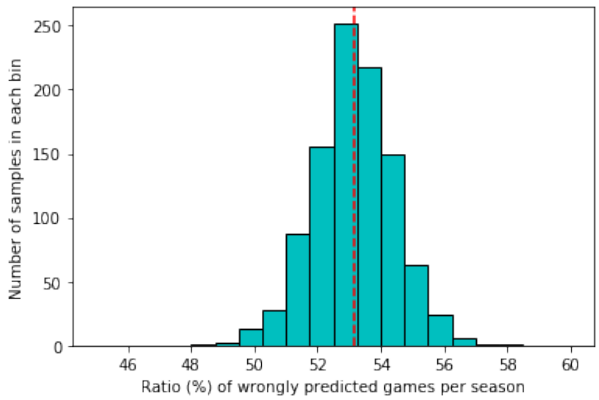

Figure 19: Histogram representing the ratio of wrongly predicted games per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 53.24.

Figures 19, 20 and 21 display results for above mentioned metric measures. As we see, results are almost equal to the ones obtained in Approach 2. Contrarily, we can see in Figure 22 how the average of predicted wins per season over 1000 different season simulations is much more similar to the ground truth number of games won, for almost every team in the NBA league, when comparing it to results obtained in Approaches 1 and 2.

In comparison to both previous approaches, results obtained are slightly better. We observe a decrease in games misprediction rate, as well as in normalized Kendall Tau distance, but contrarily there is a notable decrease in τ correlation coefficient between both obtained rankings. When comparing Figure 15 with both Figures 10 and 15, we observe

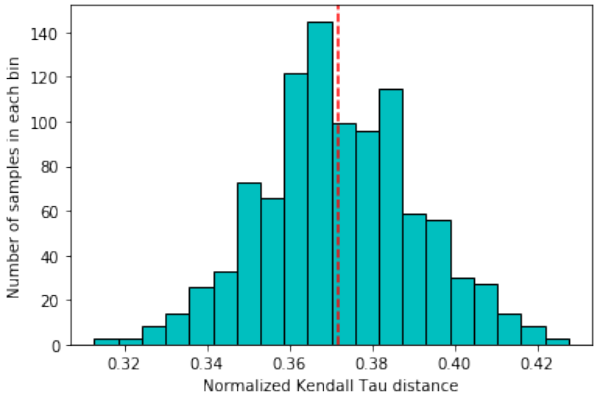

Figure 20: Histogram of predicted normalized Kendall Tau distance per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 0.37.

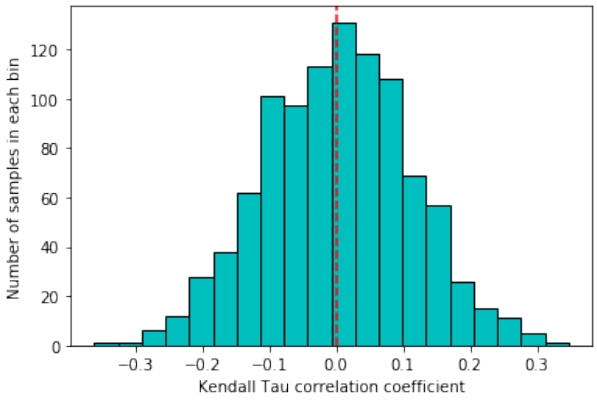

Figure 21: Histogram of predicted τ correlation coefficient per season, over 1000 different season simulations. Average mean, shown as a dashed red line, is 0.015%.

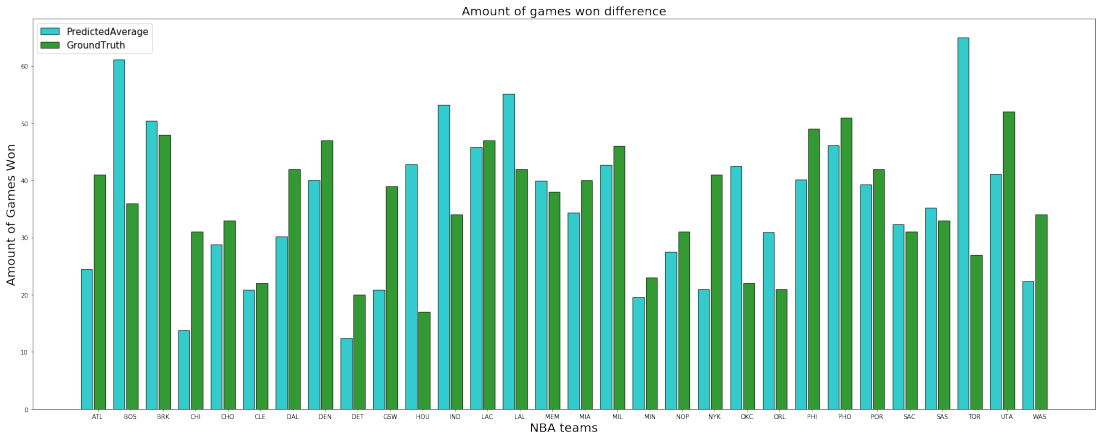

Figure 22: Histogram of predicted average wins per season, in blue, against the ground truth wins per team, in green. The average difference between both predicted and ground truth amount of win games for 2020/2021 regular season is 10.68.

a noticeable reduction in the average difference between predicted and ground truth amount of wins per team to 10.68. This noticeable improvement in this aspect is attributed to the fact that, contrarily to previously approaches, this model is able to capture players movements from one season to another. Teams skills are better represented. This happens because after inference on 2019/2020 regular season (which is performed over players), but before starting prediction model on 2020/2021 regular season, players are rearranged in their new teams according to the transfers which took place between seasons 2019/2020 and 2020/2021.

8 Conclusions

After collecting and comparing the results for each approach, global analysis of TrueSkill model can be produced. TrueSkill has been proven to work consistently in many sports (such are tennis, chess or golf), as well as for online game skills computation for matchmaking, ranking and games outcome prediction. And after this research, it also has been proven to perform consistently on basketball teams or players skills computation and game winner prediction.

Different Trueskill model implementations are developed and tested in this research. Each of them capturing different factors of the game. The first developed approach is intended to capture collective team skills, and therefore each team is considered as a single rating unit. The second developed approach is intended to capture the best players' skills, which then are grouped into teams creating a single rating unit for each team. Finally, the third developed approach is intended to also the best players' skills, but also players transfers from one team to another from a season to another. All three approaches have been proven to produce similar results.

As shown in the Results section, there is almost no difference between the first and the second approach results. Both approaches differ only in the way in which prior initialization is performed on teams, as in the first approach teams rating units are initialized using collective data statistics, whereas in the second one teams rating units are initialized as its better

three players skills aggregation. This gives an insight of how important is prior initialization for the model. As said, there is almost no difference between both approaches results, and therefore prior initialization has low, if any, effect on prediction model, as this very slight difference could also be attributed to some baseless or random fact.

The only important considerations to do regarding prior initialization are to have different skills μ initialization for players in the same team, and having large enough sigma σ values. When using players rating units for inference, players need to be initialized using different μ and σ values, to avoid having same skills for players in the same team after inference process. Secondly, it is important to have large enough σ values to provide the model with the sufficient margin to learn skills during inference. If there is low uncertainty about players or teams performance, during the inference computation such players or teams skills learning will be restricted, and inference would not affect much on players skills.

It is difficult to determine the outcome of a basketball game. There are many factors to consider. The first approach described in this research considered almost no aspect of the game. By using some prior data on teams previous season statistics, and performing inference using labeled games in 2010/2020 regular season, the model was able to correctly predict 45.21% of 2020/2021 regular season games. Despite this results are apparently quite good, they are worse than expected when comparing with TrueSkill model performance on other sports such tennis or chess

[11]. For this reason, second approach was used, in an intend to better capture teams initial skills.

But as previously discussed, prior initialization, if some conditions are satisfied, has low importance and slight or none effect on games prediction after performing inference. For this reason when using the second approach almost same results were obtained, as the model was able to correctly predict 45.27% of 2020/2021 regular season games, which again was not enough when comparing it with other sports results.

Many aspects of the game change from one season to another in NBA basketball league, and one of the most important ones is players movements from one team to another. Because of this, third model was implemented. When using players rating units instead of teams units, transfers of players were captured from one season to another. Therefore, this model was expected to perform better than both previous models. And it does, as as the model was able to correctly predict 46.76% of 2020/2021 regular season games. However with this improvement, I personally expected a better performance from this model in regard to the games prediction ratio. However, there is really an observable improve in Figure 22 with respect to both Figures 10 and 15, as it is appreciable how the predicted average amount of games win is much closer to the ground truth amount of games win for the season 2020/2021, for almost every team in the league. Therefore, the average difference between ground truth and predicted amount of games won in the season 2020/2021 is reduced from

15.65 to 10.68. This qualitative improvement in game outcome prediction is surely attributed to the fact that the model is able to capture players transfers from one season to another, as it is the main difference between the third and other two implementations.

Despite this improvement, and as mentioned before, there are still a lot aspects which, in some way, influence the outcome of a game, and that are not considered by any of the implemented models. Some such aspects are players injuries, home team advantage, teams' context, and new players' appearances in the league. Being able to capture such aspects would probably secure a better performance on games' outcome prediction, as well as in rankings distance reduction.

9 Further research

There are many conditionals of a basketball games outcome. Further research will be focused on improving performance of games' outcome prediction. Developing a model which would be able to capture, in addition to the players transfers and new rookie players appearance, the following aspects of basketball games, would definitely help in this purpose.

The first of such determinants is home team advantage. Crowd does really make a difference in almost every sport, and therefore would have an impact on games' win prediction rate.

Secondly, another aspect to consider is players injuries or non appearance in determinate games. In every developed implementation in this research, players are assumed to play every game in both inference and predictive models. This does not happen in reality, where unpredictable injuries or players missing some games because of tactical decisions may occur. This clearly does have a huge impact on a game outcome.

Finally, other important aspects in consideration would be teams' dynamics and contexts, teams' winning streak, or the fact that if its a back-to-back game for any of its teams involved. Back-to-back game is the term used to describe the second of two games which take place in two consecutive days.

References

[1] Barber, David. 2012. Factor Graphs [Chapter 4.4]. *Bayesian Reasoning and Machine Learning*. Cambridge University Press.
http://www.cs.ucl.ac.uk/staff/d.barber/brml/

[2] Barber, David. 2012. The sum-product algorithm on factor graphs [Chapter 5.1.2]. *Bayesian Reasoning and Machine Learning*. Cambridge University Press.
http://www.cs.ucl.ac.uk/staff/d.barber/brml/

[3] Barber, David. 2012. Expectation Propagation [Chapter 28.8]. *Bayesian Reasoning and Machine Learning*. Cambridge University Press.
http://www.cs.ucl.ac.uk/staff/d.barber/brml/

[4] Minka, Thomas P. 2001. *Expectation Propagation for Approximate Bayesian Inference*. Carnegie Mellon University, Statistic Department.
https://arxiv.org/pdf/1301.2294.pdf

[5] Basketball Reference Website.
https://www.basketball-reference.com/

[6] NBA National Basketball Association official website.
https://www.nba.com/stats/

[7] Lee, Heungsub. September 2015. *TrueSkill Documentation*.
https://trueskill.org/

[8] Herbrich, Ralf , Minka, Tom , and Graepel, Thore. January 2007. *TrueSkill: A Bayesian Skill Rating System*. Microsoft Research Ltd. Cambridge, UK.
https://papers.nips.cc/paper/2006/file/f44ee263952e65b3610b8ba51229d1f9-Paper.pdf

[9] Wikipedia Contributors. 2021. *Kendall tau distance*. Wikipedia.
https://en.wikipedia.org/wiki/Kendall_tau_distance

[10] Walker, David A. November 2003. *Converting Kendall's Tau For correlational Or Meta-Analytic Analyses*. Educational Research and Assessment, Northern Illinois University.
https://digitalcommons.wayne.edu/jmasm/vol2/iss2/26/

[11] Ibstedt, Julia, Radahl, Elsa, Turesson, Erik, and vande Voorde, Magdalena. 2019. *Application and Further Development of Trueskill Ranking in Sports*.
https://uu.diva-portal.org/smash/get/diva2:1322103/FULLTEXT01.pdf

[12] Moser, Jeff. May 2011. *The Math Behind TrueSkill*. https://www.moserware.com/assets/computing-your-skill/The%20Math%20Behind%20TrueSkill.pdf

[13] Koller, Daphne, and Friedman, Nir. 2010. Bayesian networks [Chapter 3]. *Probabilistic Graphical Models, Principles and Techniques*. https://djsaunde.github.io/read/books/pdfs/probabilistic%20graphical%20models.pdf