CUED - Engineering Tripos Part IIB 2018-2019 **Module Coursework**

Module	4F13	Title of report	Coursework #2: Probabilistic Ranking

Date submitted: Type text here 16/11/2018

Assessment for this module is ☐ 100% / ☐ 25% coursework

of which this assignment forms _____ %

UNDERGRADUATE STUDENTS ONLY		POST GRADUATE STUDENTS ONLY			
Candidate number:	5509E	Name:		College:	

Feedback to the student

☐ **See also comments in the text**

		Very good	**Good**	Needs improvmt
CONTENT	**Completeness, quantity of content:** Has the report covered all aspects of the lab? Has the analysis been carried out thoroughly?		✓	
	Correctness, quality of content Is the data correct? Is the analysis of the data correct? Are the conclusions correct?			✓
	Depth of understanding, quality of discussion Does the report show a good technical understanding? Have all the relevant conclusions been drawn?			✓
	Comments: Part (a): it is worth noting that it is possible that the sampler got stuck in a single mode of the distribution (which is not the case here as the distribution is unimodal). Most of the samples being centered around the mean and within a standard deviation is not really saying much when it comes to whether the sampler is working. For mixing time we were referring more to the distance between independent samples once the sampler has converged, although your definition is used in some sources. Part (b): by convergence for the Gibbs sampler we were referring to the point at which you can treat a sample as being taken from the true posterior. This is after the burn-in phase, which can be determined by eye and does not require Gelman-Rubin Diagnostics. An appropriate burn-in would be about 10 iterations. This is a different question to how many samples are actually required to get a good estimate of an expectation, which is what Gelman-Rubin Diagnostics tries to address: the sampler may have already started to sample from the target buthas not explored the entire distribution sufficiently. By initialization we were not referring to varying the hyperparameters. Part (c): as a minor point, it makes more sense to use the rows for player 1 and columns for player 2 in the tables (more consistent with standard matrix notation). Part (d): Note that it is possible to use the samples directly to compute the probability using marginals. By using the marginals you lose information as you no longer have the correlations in the joint distribution. Therefore using the joint samples is in fact better. There should also be no additional scaling problems with using the joint as you would need to re-do inference with adding players anyways as all the skills are inter-related. Part (e): It is not meaningful to compare the empirical win averages with skills as the two are different quantities. Rather, you should have computed the average probability for a player to win for each player using skills and compared those. The prior itself makes no distinction as to how many games a player has played.			
PRESENTATION	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?		✓	
	Comments:			

Overall assessment (circle grade)	A*	**A**	B	C ✓	D
Guideline standard	>75%	**65-75%**	55-65%	40-55%	<40%
Penalty for lateness:	*20% of marks per week or part week that the work is late.*				

Marker: Date:

4F13 CW1: Probabilistic Ranking

Candidate Number: 5590E

Friday Nov 16th, 2018

1 Question A

Figure 1 shows how the Gibbs sampler steps around each posterior each iteration. At a glance, it looks as though the sampler does move around the whole posterior: for Djokovic, most samples are within the first std and are centered on the mean (see Question B for justification of this).

The mixing time is defined as the number of iterations needed to reach the stationary distribution. This changes each run due to initialisation and the random nature of the samples. Consequently, we choose a burn-in time such that Burn-in time > max(Mixing Time). In figure 1 the Mixing Time looks to be around 40 iterations for the last player to reach a steady state (Nadal). A good burn-in time for this distribution is around 500 (see Question B for justification).

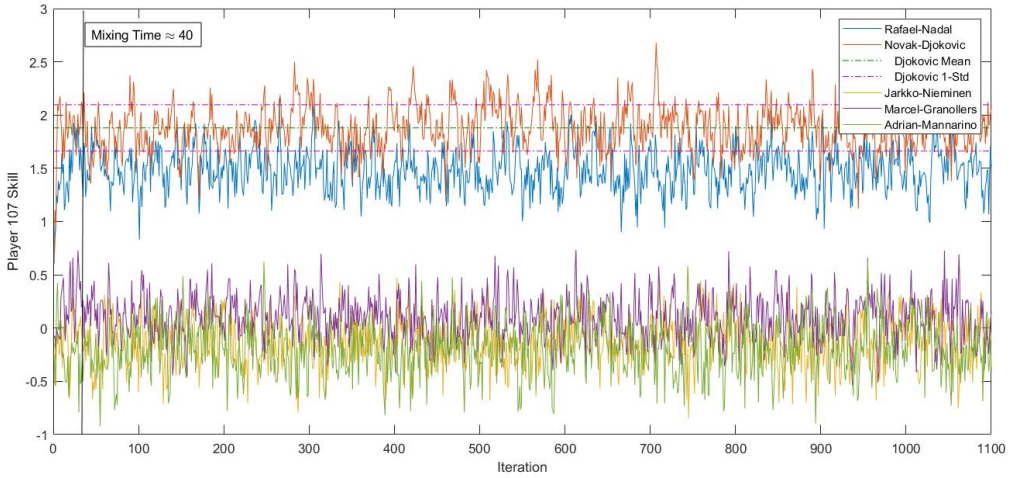

Figure 1: The sampled posterior over player skill for 5 players against the Gibbs iteration. The first standard deviation and mean of the data for Djokovic is plotted as well as a first guess for the mixing time. The mean and standard deviation for Djokovic are plotted and calculated from the last 10000 iterations of a 100000 iteration run.

Figures 2 and 3 show that the largest auto-correlation length is attributed to Djokovic at around 4.5. This is expected as he is one of the best players and has played the most games (76). Therefore, he has one of the lowest average $Var[p(w_{Djokovic}|\boldsymbol{w}_{\setminus Djokovic})]$ (or equivalently, when taking eigenvalues of the precision matrix (in the $w_1 - w_2$ direction), he has the highest average precision) causing the joint, $p(w_1, w_2)$[1], to squash: increasing the time to traverse the length of the distribution. Consequently, only every 5th sample (or more) from the Gibbs sequence is independent of one another. We can use this to thin/sub-sample the data to get an independent data-set that more accurately measures the true joint distribution's statistics[2].

[1] Note that the joint can be written as $p(w_1, w_2)$ or $p(w_1, t)$ since t is a function of w_2.

[2] Thinning is not strictly necessary since, given enough samples, the joint distribution will tend to the true distribution anyway [1]

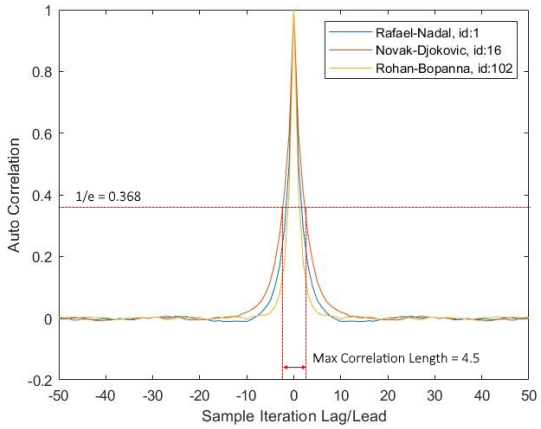

Figure 2: A plot of the auto-correlation coefficients of three different players using 100,000 iterations. The auto- correlation length, L_{corr}, for Djokovic is given by the width of the peak at 1/e of the maximum value.

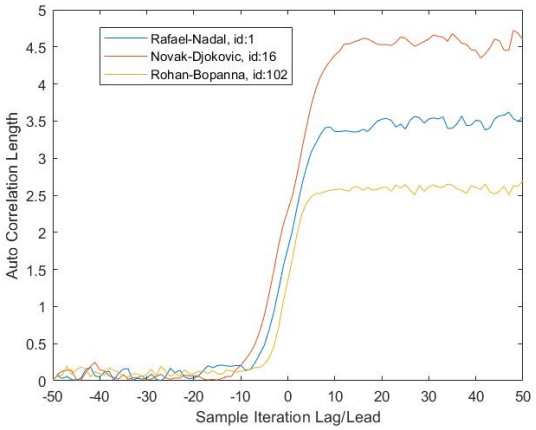

Figure 3: An equivalent measure of the auto-correlation length is given by its integral over 100,000. The length can be read from the right hand side of the figure. Djokovic has the largest correlation length at ≈ 4.5.

```
m(p) = t'*(    ( (G(:,1) - p) == 0 )  -  ( (G(:, 2) - p) == 0 )    );

iS(G(g,1), G(g,1)) = iS(G(g,1), G(g,1)) + 1;
iS(G(g,2), G(g,2)) = iS(G(g,2), G(g,2)) + 1;
iS(G(g,1), G(g,2)) = iS(G(g,1), G(g,2)) - 1;
iS(G(g,2), G(g,1)) = iS(G(g,2), G(g,1)) - 1;
```

Figure 4: Two code snippets used to generate the Mean Vector and Covariance Matrix for jointly sampling skills given performance differences in the Gibbs sampling algorithm.

2 Question B

Gibbs sampling alternates between sampling from the skills given performances, $p(\boldsymbol{w}|\boldsymbol{t},\boldsymbol{y})$, and performances given skills, $p(\boldsymbol{t}|\boldsymbol{w},\boldsymbol{y})$. Over many iterations we converge to the joint distribution, $p(\boldsymbol{w},\boldsymbol{t}|\boldsymbol{y})$, because each sample is drawn from a conditional (i.e. some slice of the joint) which effectively "pulls" the samples towards the mean of the joint. The marginal, $p(\boldsymbol{w})$, can then be estimated by averaging over samples from $p(\boldsymbol{w}|\boldsymbol{t},\boldsymbol{y})$ at different performances. Figure 5 gives an illustration of what these distributions look like for one player.

Gibbs convergence occurs when the samples have the same values as if they were drawn from the true joint distribution, i.e. their statistics are stationary. If the joint has converged then the marginals will also have converged and since we are only interested in the marginal skill we can assess its convergence using Gelman-Rubin Diagnostics [2]:

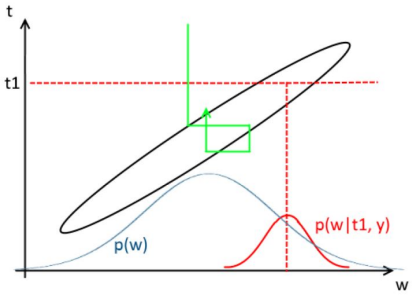

Figure 5: An illustration of Gibbs sampling showing: a sample trajectory alternating between fixing w and t (green); the marginal (blue); the conditional posterior over w (red) at the previously sampled performance, t1; and the joint distribution (black).

$$B = \frac{N}{M-1}\sum_{m=1}^{M}(\hat{\mu}_m - \hat{\mu})^2, \qquad W = \frac{1}{M}\sum_{m=1}^{M}\hat{\sigma}_m^2 \tag{1}$$

$$Pooled\,Variance,\ \hat{V} = \frac{N-1}{N}W + \frac{M+1}{NM}B \tag{2}$$

$$PSRF = \frac{\hat{V}}{W} \tag{3}$$

Where B is the between-chain variance, measured as the sum of squared errors of the chain mean $\hat{\mu}_m$ and the average chain mean $\hat{\mu}$, and W is the within-chain variance. When converging to a stationary distribution B will tend to 0 and the potential scale reduction factor (PSRF) will tend to 1. Figure 6 shows the PSRF plotted for three different players and shows that it will take an infinite number of iterations to truly converge, however, after 500 iterations it comes fairly close to the true posterior[3].

Different pseudo-random number seeds do not affect whether the sampler will converge, only the starting location, see figure 6.

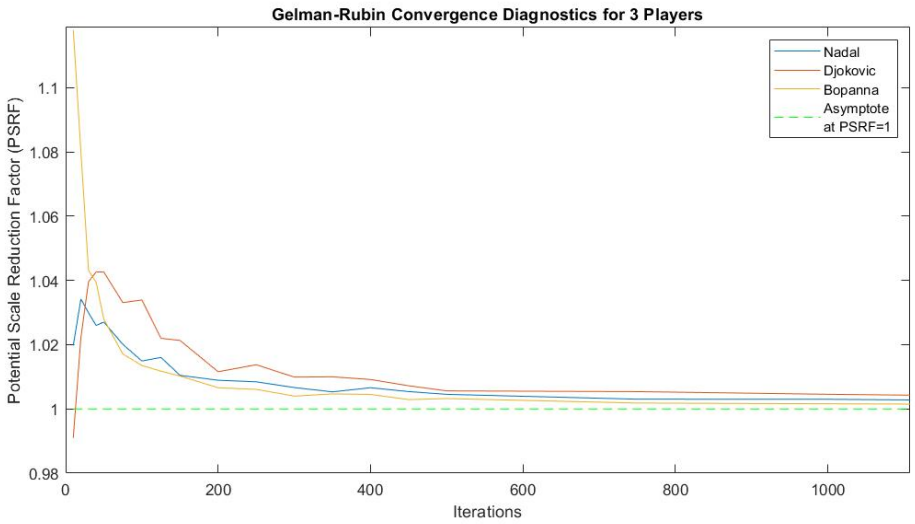

Figure 6: The PSRF plotted at different iteration numbers. At each iteration, 200 randomly initialized chains were generated and compared.

For the message passing algorithm (MPA), convergence is "usually" well approximated when there is no significant difference in belief update [4], this can be measured in terms of absolute error from the asymptotic values. The MPA converges directly to the marginal skills because it is inferring the moments for each player.

[3]Calculating the between-chain variance and within-chain variance is effectively checking whether the time averages are equal to the ensemble averages, i.e. whether the distribution is ergodic/stationary. This is one of the two conditions required for the gibbs algorithm to sample from the correct distribution. The other being invariance of the marginals [3].

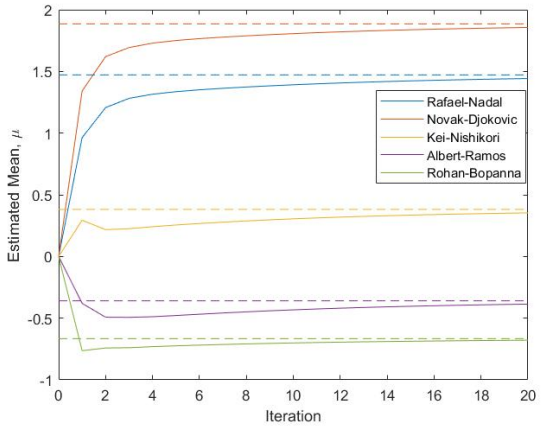

Figure 7: The approximate means of players as the number of iterations increases. The asymptotic means are plotted with dashed lines for comparison.

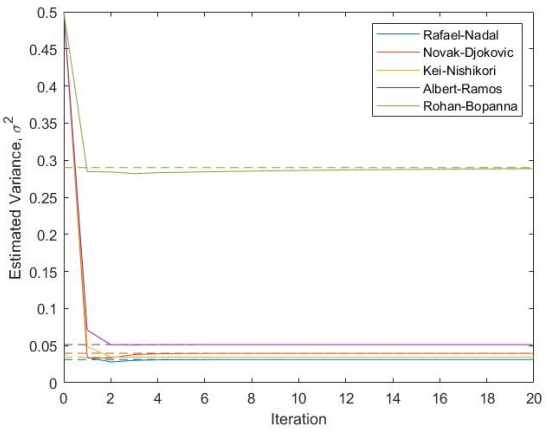

Figure 8: The approximate variances of players as the number of iterations increases. The asymptotic variances are plotted with dashed lines for comparison.

After 20 iterations the largest error in the means is $\approx 2.8 \times 10^{-2}$ (Nishikori). Figures 7 and 8 show that the variance errors decrease much more rapidly than the means and that 20 is a reasonable number of iterations to say the MPA has converged.

```
% initialize matrices of game to skill messages - means and precisions
Mgs = param1.*ones(NumGames,2);        Pgs = param2.*ones(NumGames,2);
```

Figure 9: Initialising the game to skill messages to be changed using param1 and param2. Originally param1 = param2 = 0.

When initialising the game-to-skill messages differently (Figure 9), the marginal skill moments converge to the same values. When initialising the prior variances differently, because the variance is only included in the difference between natural means, it doesn't affect the ranking order, only the spread. I.e. larger priors increase the mean skill of better players, and decrease the those of worse players whilst preserving the order and relative difference between player skills.

3 Question C

$p(w_1 > w_2)$		Player 1 (P1)			
		Djokovic	Nadal	Federer	Murray
Player 2 (P2)	Djokovic	0.5000	0.0602	0.0911	0.0147
	Nadal	0.9398	0.5000	0.5728	0.2335
	Federer	0.9089	0.4272	0.5000	0.1892
	Murray	0.9853	0.7665	0.8108	0.5000

Table 1: The probability that the skill of player 1 is higher than player 2, $p(w_1 > w_2)$, for the top 4 players using the MPA, where w is the player skill.

$p(y = 1 \mid w_1, w_2)$		Player 1 (P1)			
		Djokovic	Nadal	Federer	Murray
Player 2 (P2)	Djokovic	0.5000	0.3446	0.3620	0.2802
	Nadal	0.6554	0.5000	0.5184	0.4269
	Federer	0.6380	0.4816	0.5000	0.4091
	Murray	0.7198	0.5731	0.5909	0.5000

Table 2: The probability that player 1 wins the match given both player's skills $p(y = 1|w_1, w_2)$, where y is the probability that player 1 wins ($y = 1$) or loses ($y = 0$).

Skill is a latent variable associated with each player, whereas the probability of a player winning is given by $p(t > 0|w_1, w_2)$ where t is the performance difference. The performance difference takes into account the additional uncertainty of performing on the day by introducing a prior with mean 0 and variance 1 (figure 10). This has the effect of pulling the probability of one player winning over the other closer to 0.5, even if there is a large true skill difference between them. This is shown by the probabilities in table 2 being closer to 0.5 than their corresponding entries in table 1.

Note that in tables 2 and 1 the probability of Dokovic's skill being greater than Nadal's is lower compared to that of Federers. This is because the tables are ordered from left to right according to the ATP rankings, whereas the MPA predicts the skill of Federer to be higher than Nadals.

```
pmean = Ms_top(row) - Ms_top(col);
variance = 1/(Ps_top(row)) + 1/(Ps_top(col));
skill_matrix(row,col) = normcdf((0-pmean)/sqrt(variance));
t_variance = variance + 1; % + performance inconsistancy
result_matrix(row,col) = normcdf((0-pmean)/sqrt(t_variance));
```

Figure 10: Code used to generate the data in tables 1 and 2 .

4 Question D

$p(w_1 > w_2)$		Player 1 (P1)	
		Djokovic	Nadal
Player 2 (P2)	Djokovic	0.5000	0.0764
	Nadal	0.9236	0.5000

Table 3: A table showing the probability of the skill of player 1 being greater than that of player 2. Calculated by first computing the marginal skill for each player.

$p(w_1 > w_2)$		Player 1 (P1)	
		Djokovic	Nadal
Player 2 (P2)	Djokovic	0.5000	0.0480
	Nadal	0.9520	0.5000

Table 4: A table showing the probability of the skill of player 1 being greater than that of player 2. Calculated by directly inferring this from the joint samples (figure 11).

Computing the marginal skill first (table 3) is a generative approach because it requires both the marginal moments to be calculated, whereas using joint samples calculates the conditional probability of $w_1 > w_2$ given sampled observations and is therefore discriminative. Using joint samples overestimates the probability of $p(w_{Djokovic} > w_{Nadal})$ because information about the variances of the posteriors is lost by making each sample binary.

Computing the marginal skills also has the advantage of having standalone statistics, whereas computing the joint requires comparison between players. Therefore, if a new player is added they will need to be compared with every other player rather than just computing their marginal skill. This has problems with scaling. Thus, computing the marginal skills first is generally a better way to compare players.

$p(w_1 > w_2)$		Player 1 (P1)			
		Djokovic	Nadal	Federer	Murray
Player 2 (P2)	Djokovic	0.5000	0.0764	0.1077	0.0216
	Nadal	0.9236	0.5000	0.5644	0.2501
	Federer	0.8923	0.4356	0.5000	0.2083
	Murray	0.9784	0.7499	0.7917	0.5000

Table 5: Probabilities of player 1's skill being greater than player 2's computed by first calculating the marginal from Gibbs sampling. For comparison with table 1.

Comparing table 5 with 1 we see that the table 5 predicts slightly less extreme probabilities, i.e. they are closer to 0.5. The reasons for this are not clear.

```
% Count samples where skill(p1) > skill(p2). If same add 0.5
sampled_wins = (sW(col, :) > sW(row, :)) + 0.5*(sW(row, :) == sW(col, :));
joint_matrix(row,col) = mean(sampled_wins);
```

Figure 11: Code used to directly infer which player has the greater skill in 4 from the joint samples. The code used to compute the marginals first is the same as that in figure 10.

5 Question E

The rankings based on empirical game averages (figure 12) are substantially different from those in figures 13 and 14. Figure 12 does not take into account the skills of each player or the number of games played. This causes there to be a few players with "0 Skill" since they won 0 games.

Conversely, figures 13 and 14 assign every player a skill because they take into account the skill of the people that they played allowing those who lost all their games, but to higher ranked players, to be ranked higher. The number of games played is taken into account by incorporating a prior over each player's skill. Additionally, because the better ranked players tend to play more games and play against other better players, their skill is boosted significantly compared to those of lower skill. This skews the histograms left.

Figures 13 and 14 are very similar and are both attempting to approximate the true marginal distributions. There are discrepancies in both, possibly because of the method with which they approximate the true marginals: Gibbs averages over many samples, whereas message passing uses a moment matching approximation.

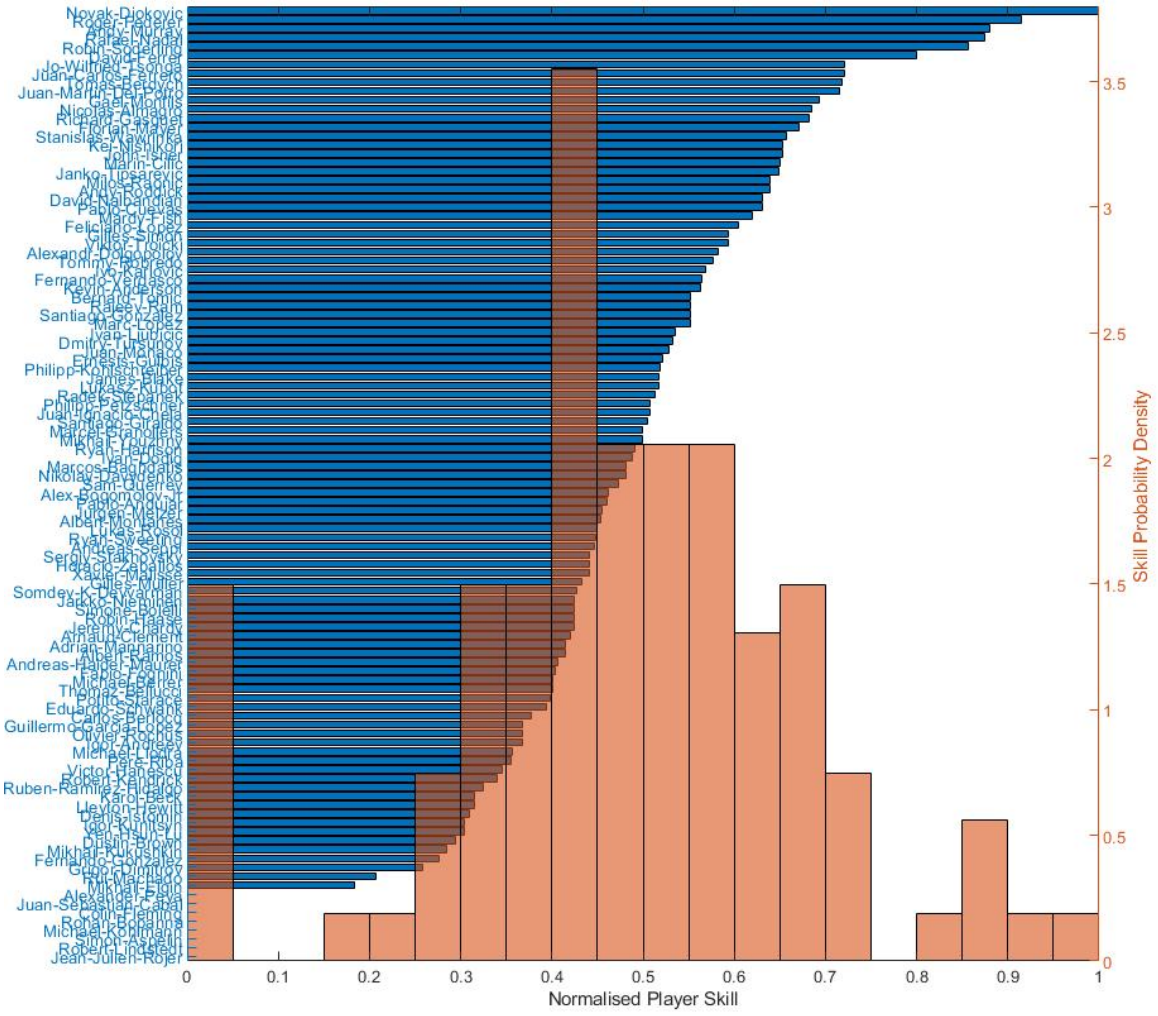

Figure 12: Ranking and Normalised Skill level based on empirical game outcome averages. A histogram binning the skills of similar players is shown overlaid.

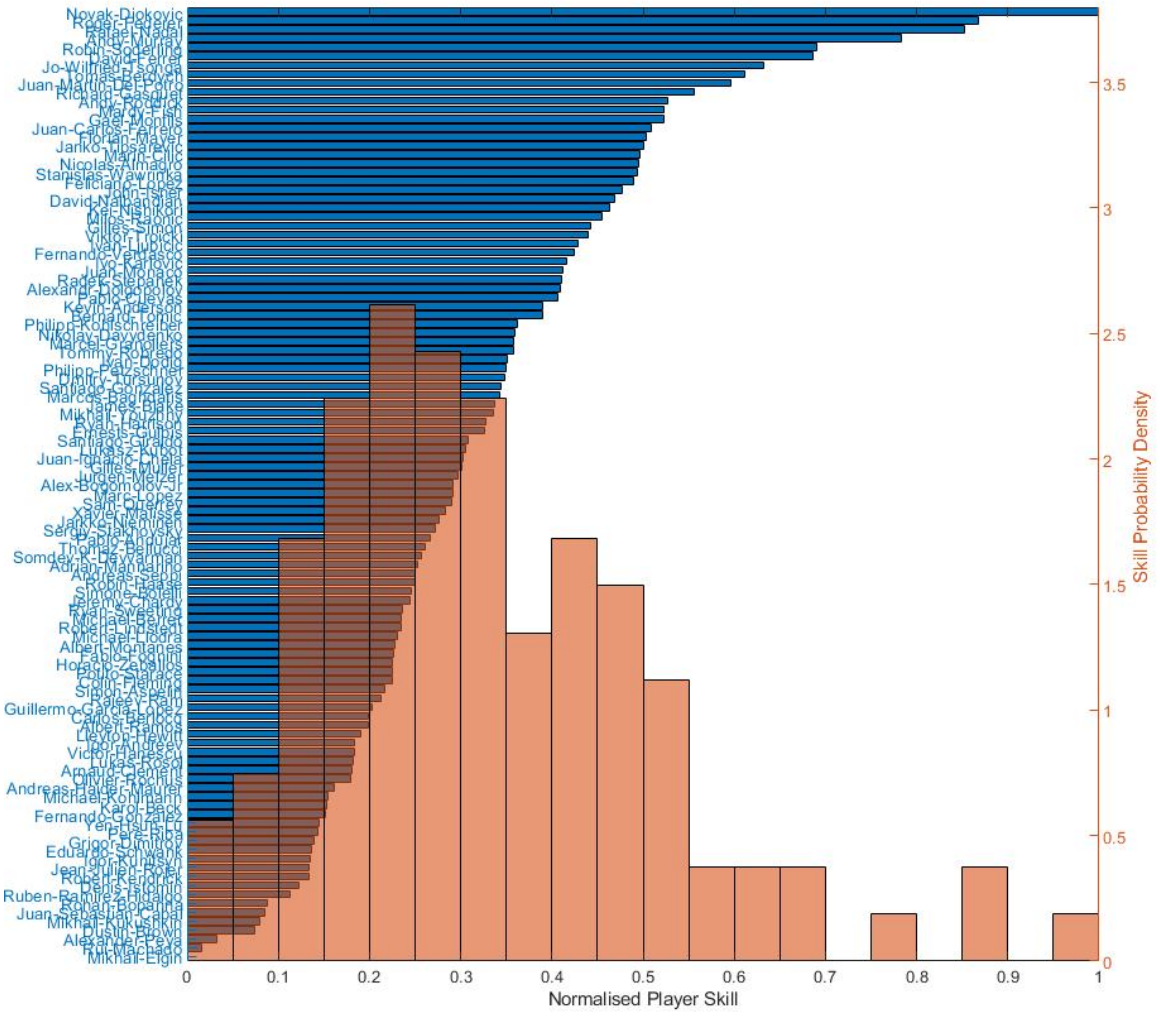

Figure 13: Ranking and Normalised Skill level based on mean predictions from gibbs sampling. A histogram binning the skills of similar players is shown overlaid.

References

[1] William A. Link Mitchell J. Eaton (2011). *On thinning of chains in MCMC.* Methods Ecol. Evol., 3(1), pp. 112-115

[2] Andrew Gelman and Donald B. Rubin. (1992). *Inference from Iterative Simulation Using Multiple Sequences.* Statistical Science, 7(4), pp. 457-472.

[3] Christopher M. Bishop (2006). *Pattern Recognition and Machine Learning* Ch.11, pp.543-544.

[4] Murphy, K. P., Weiss, Y., and Jordan, M. I. (1999). *Loopy belief propagation for approximate inference: An empirical study.* Proceedings of Uncertainty in AI, 9, pp. 467–475.

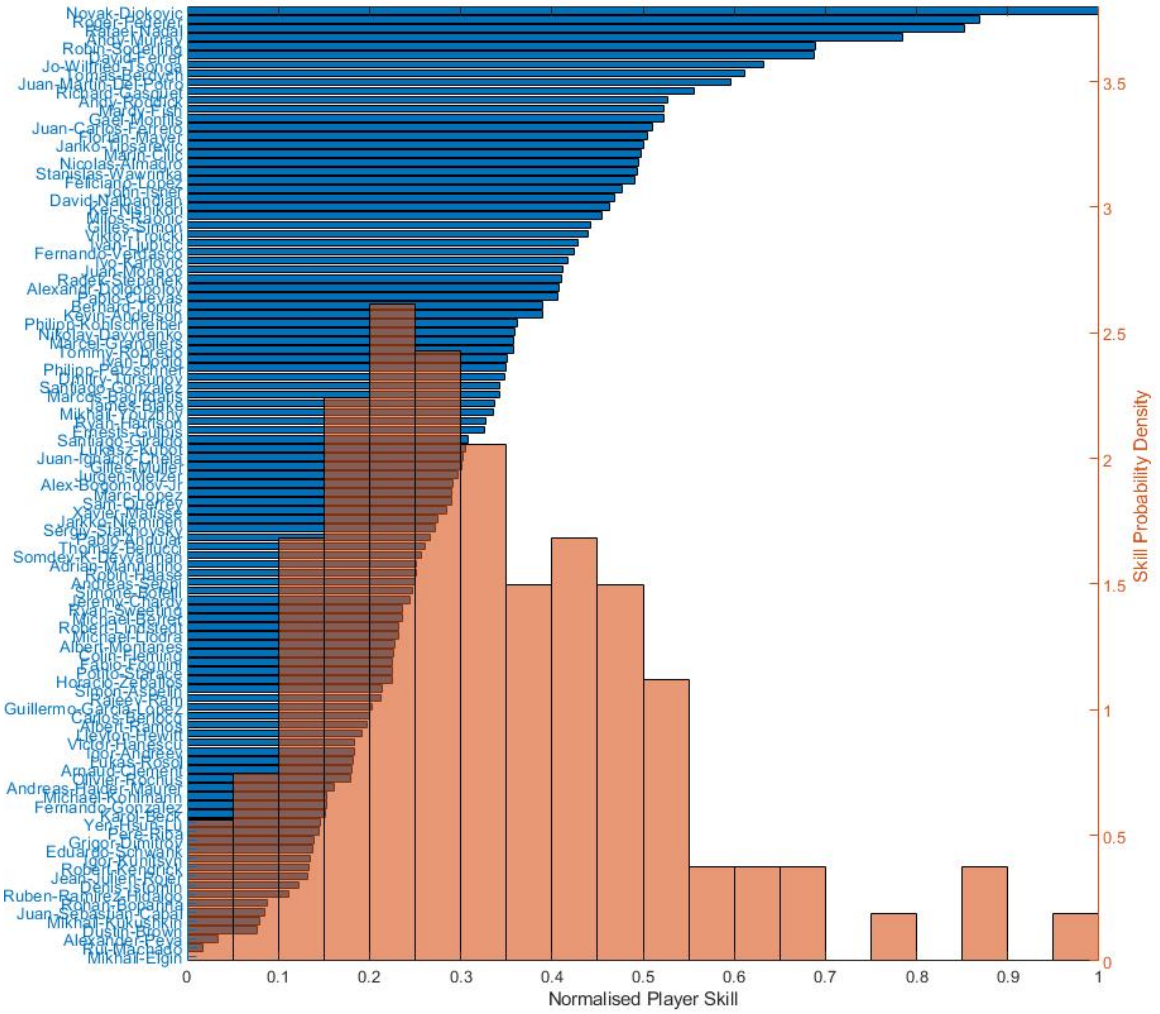

Figure 14: Ranking and Normalised Skill level mean predictions from the message passing algorithm. A histogram binning the skills of similar players is shown overlaid.

```
for p = 1:NumPlayers
    wins(p) = sum(   ( (G(:,1) - p) == 0 )   );
    games(p) = sum(   ( (G(:,1) - p) == 0 )  +  ( (G(:, 2) - p) == 0 )   );
end
frac_wins = wins./games;
```

Figure 15: Code used to calculate the empirical game outcome averages in figure 12.