CUED - Engineering Tripos Part IIB 2019-2020 **Module Coursework**

Module	4F13	Title of report	Probabilistic Ranking

Date submitted: 22nd of November 2019

Assessment for this module is ☐ 100% / ☐ 25% coursework of which this assignment forms _____ %

UNDERGRADUATE STUDENTS ONLY		POST GRADUATE STUDENTS ONLY			
Candidate number:	5741B	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:			
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:

Probabilistic Ranking

Part A

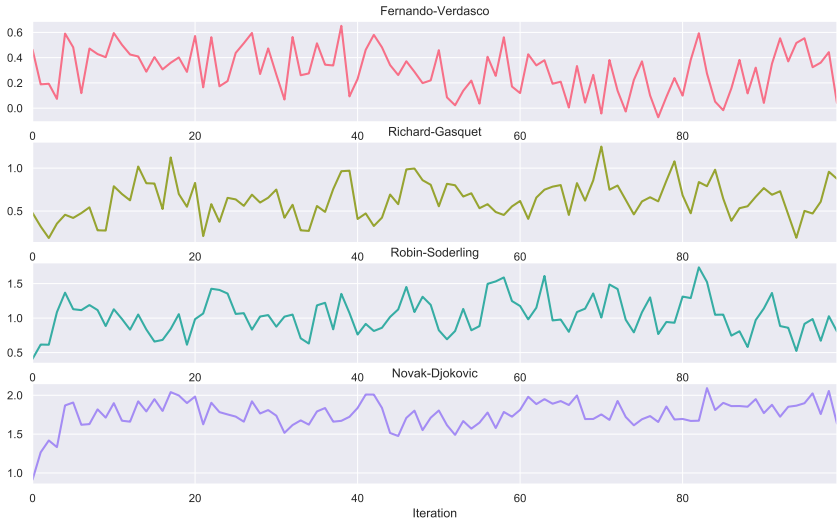


Figure 1: Skill samples from the Gibbs sampling process.

Figure 1 shows skill samples from the Gibbs sampling process for 4 players. As expected, an initial transient can be seen in some of those as the samples move into a high probability region of the distribution.

Gibbs sampling only gives an unbiased estimate once in steady-state. Hence, we want to reject the initial samples that are not representative of the steady-state distribution. To find a suitable burn-in time, for each sample $\mathbf{w}^{(i)}$ we can calculate the joint probability of the observed game outcomes $\mathbf{y}$ and the sampled skills:

$$\log p(\mathbf{y}, \mathbf{w}^{(i)}) = \log \underbrace{p(\mathbf{w}^{(i)})}_{\text{Skill prior}} + \log \underbrace{p(\mathbf{y}|\mathbf{w}^{(i)})}_{\text{Likelihood}} \tag{1}$$

$$= \log \mathcal{N}(\mathbf{w}^{(i)}; \mathbf{0}, I) + \sum_{p_w, p_l \in \mathcal{G}} \log p(y = 1 | \mathbf{w}_{p_w}, \mathbf{w}_{p_l}) \tag{2}$$

$$= \log \mathcal{N}(\mathbf{w}^{(i)}; \mathbf{0}, I) + \sum_{p_w, p_l \in \mathcal{G}} \log \Phi(\mathbf{w}_{p_w} - \mathbf{w}_{p_l}) \tag{3}$$

where $\mathcal{G}$ is the collection of games with p_w and p_l being the winning and losing players respectively.

Figure 2 shows the joint log-probability against iteration for the process. Seemingly, after more than 10 steps the process has moved into a high probability region. As a safety margin, we can reject the initial 100 samples.

Another consideration is how many samples do we need for an accurate estimate of $\mathbb{E}_{\mathbf{w} \sim p(\mathbf{w}|\mathbf{y})}[f(\mathbf{w})]$. The variance of the estimate is:

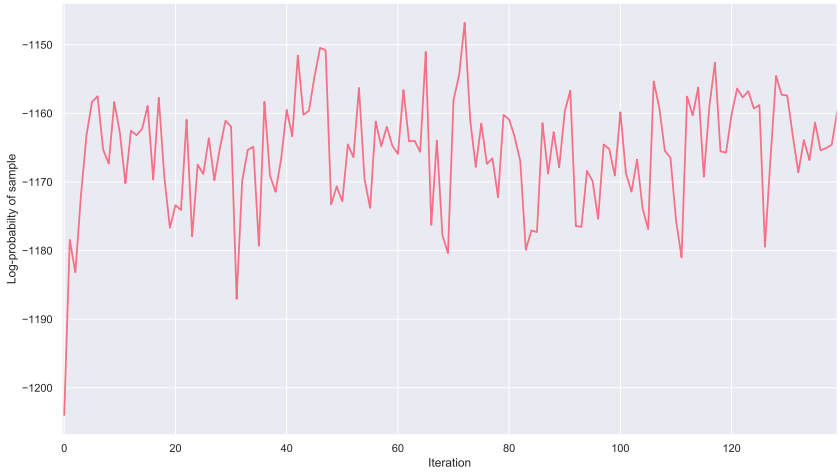


Figure 2: Log joint probability of the sampled skills and observed game outcomes.

$$\mathbb{V}\left[\frac{1}{T}\sum_{i=1}^{T} f\left(\mathbf{w}^{(i)}\right)\right] = \mathbb{E}\left[\left(\frac{1}{T}\sum_{t=1}^{T} f\left(\mathbf{w}^{(i)}\right)\right)^2\right] \tag{4}$$

$$= \frac{\mathbb{V}[F(x)]}{T}\left(1 + 2\sum_{\tau=1}^{T-1}\left(1 - \frac{\tau}{T}\right)\frac{C_\tau}{C_0}\right) \tag{5}$$

$$\approx \frac{\mathbb{V}[F(x)]}{T}\left(1 + 2\sum_{\tau=1}^{\infty}\frac{C_\tau}{C_0}\right) \qquad \text{(for sufficiently large } T\text{)} \tag{6}$$

where $C_\tau = \mathbb{E}\left[f\left(\mathbf{w}^{(i)}\right) f\left(\mathbf{w}^{(i+\tau)}\right)\right]$ and $\frac{C_\tau}{C_0}$ is the auto-correlation. Hence, the variance of the estimate is inversely proportional to the number of samples, but is larger by a factor of $\left(1 + 2\sum_{\tau=1}^{\infty}\frac{C_\tau}{C_0}\right)$ compared to an *iid* Monte-Carlo estimate.

Figure 3 shows the auto-correlation for skill samples. The auto-correlation seems to decay to 0 for all players for iteration lag $\tau < 10$. For estimating the mean, $\left(1 + 2\sum_{\tau=1}^{\infty}\frac{C_\tau}{C_0}\right)$ is equal to 1 plus the area under the auto-correlation function. we can approximate $\left(1 + 2\sum_{\tau=1}^{\infty}\frac{C_\tau}{C_0}\right) \approx 10$. Then, 10000 samples would give an estimate variance $\approx 10^{-3}$ times smaller than the variance of the skill posterior.

Part B

In Gibbs sampling, the goal is to obtain samples from the distribution $p(\mathbf{w}|\mathbf{y})$ in order to get unbiased estimates of quantities of interest of the form $\mathbb{E}_{\mathbf{w}\sim p(\mathbf{w}|\mathbf{y})}[f(\mathbf{w})]$. [1] Hence, convergence is achieved when the sampling process has approximately settled into a steady-state, and the samples become representative of the true distribution $p(\mathbf{w}|\mathbf{y})$. It is expected that the post-convergence samples will oscillate back and forth as they traverse the support of the distribution. As such, it is challenging to estimate the exact time the sampling process has converged.

In the message passing (MP) algorithm, the marginals are approximated as Gaussians. With each iteration of the algorithm, we hope that the estimate of the parameters of those Gaussians become

[1] For some applications, it might be desirable for the samples to be independent. *Thinning* is one way to obtain less correlated samples. However, as this is not necessary for the applications considered in this report, thinning is not used.

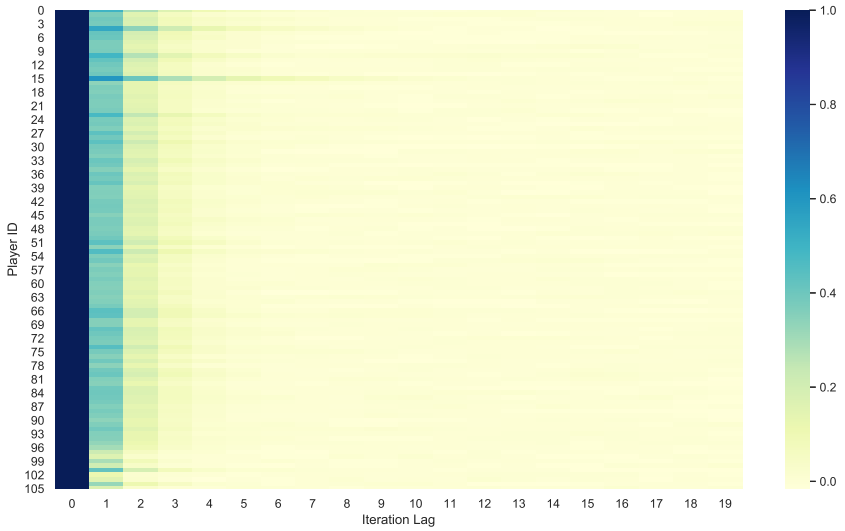


Figure 3: Auto-correlation for skill samples for all the players against iteration lag.

better. As such, we can judge convergence by whether our estimates are changing significantly; we can say that the process has converged when:

$$|\mu_j^{(i+1)} - \mu_j^{(i)}| < \epsilon \qquad \text{and} \qquad |\lambda_j^{(i+1)} - \lambda_j^{(i)}| < \epsilon \qquad \forall j$$

for some small constant ϵ, where $\mu_j^{(i)}$ and $\lambda_j^{(i)}$ are the estimates of the mean and precision of player j at iteration i.

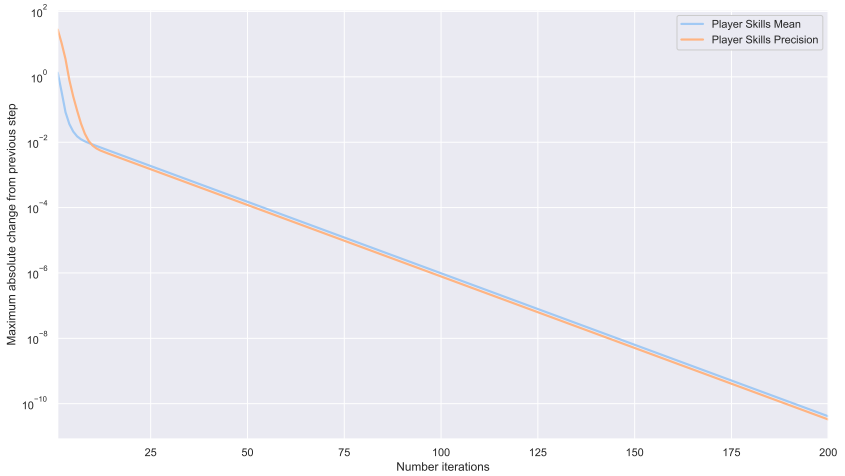


Figure 4: Maximum change in mean and precision between consecutive iterations of the message passing algorithm.

Figure 4 shows $\max_j |\mu_j^{(i+1)} - \mu_j^{(i)}|$ and $\max_j |\lambda_j^{(i+1)} - \lambda_j^{(i)}|$ against the iteration of the MP algorithm. After 100 iterations, the change in the estimated parameters is less than 10^{-6}, and the process can be said to have converged.

Part C

Once the posterior skills distribution has been approximated with independent Gaussians using the MP algorithm, the probability that a player p_1 has better skill than player p_2 can be calculated:

$$p(\mathbf{w}_{p_1} > \mathbf{w}_{p_2}) = p(\mathbf{w}_{p_2} - \mathbf{w}_{p_1} < 0) = \Phi(\frac{\mu_{p_1} - \mu_{p_2}}{\sqrt{\lambda_{p_1}^{-1} + \lambda_{p_1}^{-1}}}) \tag{7}$$

Similarly, the probability that player p_1 wins against p_2 is given by:

$$p(\mathbf{w}_{p_1} - \mathbf{w}_{p_2} + n > 0) = \Phi(\frac{\mu_{p_1} - \mu_{p_2}}{\sqrt{\lambda_{p_1}^{-1} + \lambda_{p_1}^{-1} + 1}}) \tag{8}$$

where $n \sim \mathcal{N}(0, 1)$ is the performance noise.

```
prob_higher_skill = scipy.stats.norm.cdf(0, player_2_mean - player_1_mean,
                                         (player_1_var + player_2_var)**0.5)
```

Listing 1: Command for calculating probability of player 1 having higher skill than player 2, where `player_1_mean`, `player_2_mean`, `player_1_var` and `player_2_var` are the parameters estimated using the message passing algorithm.

```
prob_wins = scipy.stats.norm.cdf(0, player_2_mean - player_1_mean,
                                 (player_1_var + player_2_var + 1.0)**0.5)
```

Listing 2: Command for calculating probability of player 1 winning against player 2 as estimated using the message passing algorithm.

Table 1 shows the probability of a player having higher skill than the other in ATP 2011 top 4. Table 2 shows the probability of a player winning against the other. I can be seen that a player with higher skill is always expected to win. However, it can also be seen that we are always less confident about a player winning or losing than we are about them having a higher or lower skill. This is expected as there is always the added performance noise when determining the game outcome, which increases our uncertainty about the outcome.

		Player 2			
		Novak Djokovic	Rafael Nadal	Roger Federer	Andy Murray
Player 1	Novak Djokovic	-	0.940	0.909	0.985
	Rafael Nadal	0.0602	-	0.427	0.767
	Roger Federer	0.0911	0.573	-	0.811
	Andy Murray	0.0147	0.233	0.189	-

Table 1: Probability that player 1 has a higher skill than player 2, as computed using the message passing algorithm after 200 steps.

Part D

Figure 5 shows the heatmap of the Gibbs samples for Djokovic and Nadal. We can again estimate $p(w_{\texttt{Djokovic}} > w_{\texttt{Nadal}})$ from the samples in several different ways. (1) by fitting a Gaussian to each marginal, and calculating the probability assuming $w_{\texttt{Djokovic}}$ and $w_{\texttt{Nadal}}$ are independent, (2) by fitting a joint Gaussian (i.e. allowing covariance), and (3) directly from the samples. Estimates of $p(w_{\texttt{Djokovic}} > w_{\texttt{Nadal}})$ for the 3 methods are shown in table 3.

We expect method (2) to yield a better estimate than method (1), since the latter ignores the covariance between the skills, which can be seen to be non-zero in figure 5. Method (3) on the other

		Player 2			
		Novak Djokovic	Rafael Nadal	Roger Federer	Andy Murray
Player 1	Novak Djokovic	-	0.655	0.638	0.720
	Rafael Nadal	0.345	-	0.482	0.573
	Roger Federer	0.362	0.518	-	0.591
	Andy Murray	0.280	0.427	0.409	-

Table 2: Probability that player 1 wins against player 2, as computed using the skill distributions from the message passing algorithm after 200 steps.

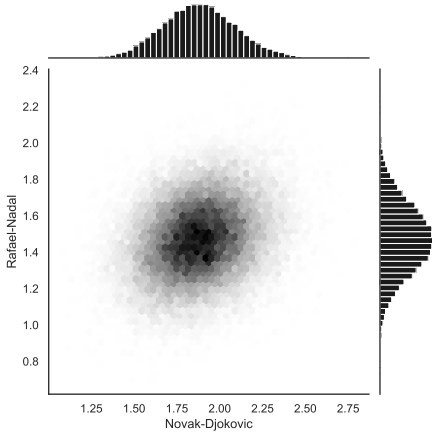


Figure 5: Heatmap of the Gibbs samples for skills for Djokovic and Nadal, with a histogram of the marginal samples distribution for these players.

Method (1)	Method (2)	Method (3)
0.92250	0.94801	0.94850

Table 3: Estimates of $p(w_{\texttt{Djokovic}} > w_{\texttt{Nadal}})$ for the 3 methods described in section D.

hand, estimates $p(w_{\texttt{Djokovic}} > w_{\texttt{Nadal}})$ directly as:

$$p(w_{\texttt{Djokovic}} > w_{\texttt{Nadal}}) = \mathbb{E}_{\mathbf{w}\sim p(\mathbf{w}|\mathbf{y})}[\mathbb{1}_{w_{\texttt{Djokovic}} > w_{\texttt{Nadal}}}] \approx \frac{1}{N}\sum_i \mathbb{1}_{w^{(i)}_{\texttt{Djokovic}} > w^{(i)}_{\texttt{Nadal}}} \tag{9}$$

Since method (3) doesn't assume anything about the distribution, it is expected to be the most unbiased out of the three, especially as the posterior skill distribution is not necessarily Gaussian. Hence, method (3) likely gives the best estimate.However, as the distribution *looks* approximately Gaussian, it is not surprising to see that estimates using method (2) and (3) are very similar.

It is important to note, however, that when comparing many players it is more computationally efficient to first estimate the parameters of the marginals, and then use those for further inference.

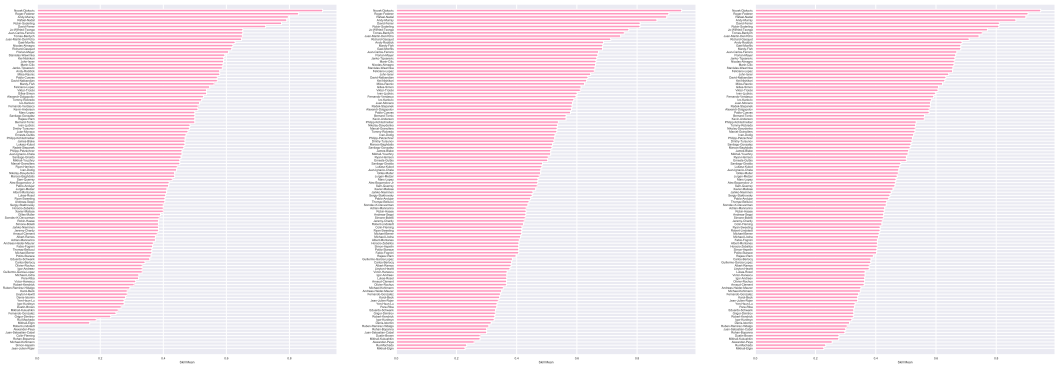
Table 4 shows the probability of a player having a higher skill using method (3) (it is the equivalent of table 1 for Gibbs sampling). The estimates are

Part E

Figure 6 shows the ranking of players using an estimate of expected probability of winning against a randomly (uniformly) chosen player. Full-sized versions are available in the appendix. These demonstrate an issue with using the empirical outcome averages: some players cannot be compared

		Player 2			
		Novak Djokovic	Rafael Nadal	Roger Federer	Andy Murray
Player 1	Novak Djokovic	-	0.949	0.917	0.988
	Rafael Nadal	0.0511	-	0.425	0.785
	Roger Federer	0.0828	0.575	-	0.813
	Andy Murray	0.0119	0.215	0.187	-

Table 4: Probability that player 1 has a higher skill than player 2, as computed directly from the Gibbs samples.



(a) Ranking using empirical game outcome averages.

(b) Ranking using expected outcome - MP.

(c) Ranking using expected outcome - Gibbs.

Figure 6: Ranking of tennis players using game outcome averages and expected skill.

if they won the same fraction of games. This is, for instance, the case for players who lost all of their games. This issue is alleviated when using estimates of expected outcome averages using the posterior over skills.

Figure 7 shows the joint scatter plots of the expected outcome averages. Clearly, MP and Gibbs give very similar estimates, and both are quite different from the empirical outcome averages. An especially interesting phenomenon occurs for players who lost of their games, some of which, such as Simon Aspelin, are ranked higher than other players who did win some games when using TrueSkill. This reflects the fact that they played against challenging opponents, whereas the players at the very bottom of the table likely lost against players with relatively low skills.

Figures 11 and 12 show the rankings using the expected skill. These are different from the ones based on expected game outcomes, due to the difference in variances of the estimates. Note that variances for some players are larger than the others. It is an interesting research question to consider what choice of two players for the next game is expected to reduce the uncertainty in the ranking the most.

Word Count: 999

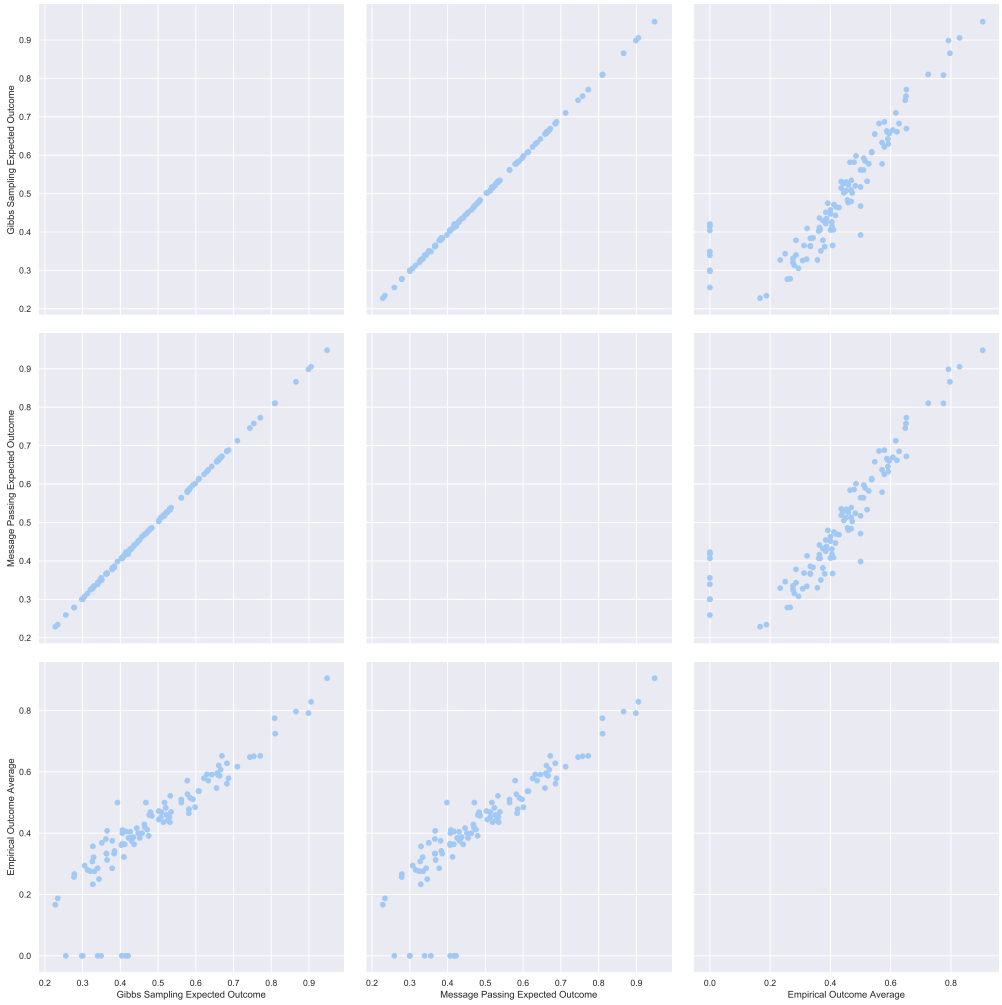


Figure 7: Joint plot of the rankings using expected outcome against a random player.

Appendix

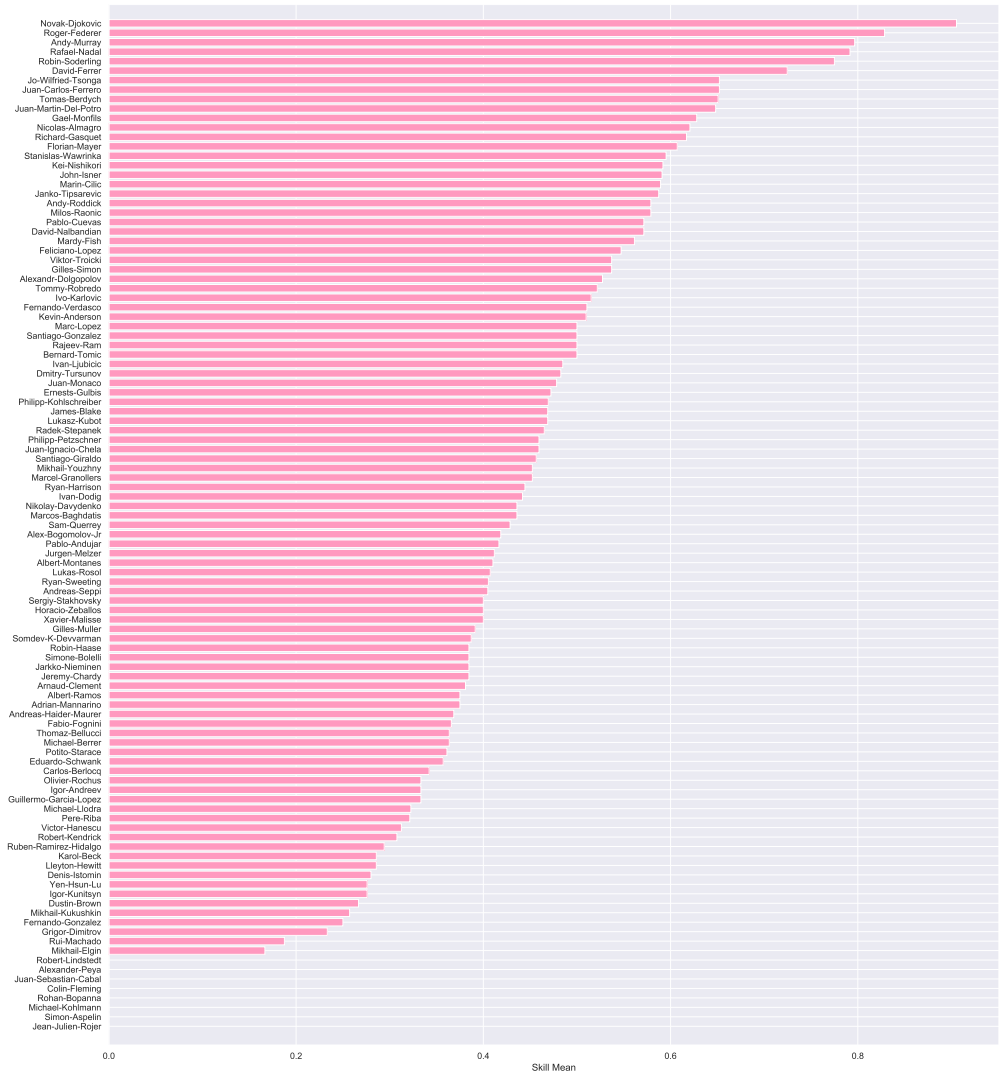


Figure 8: Ranking using empirical game outcome averages.

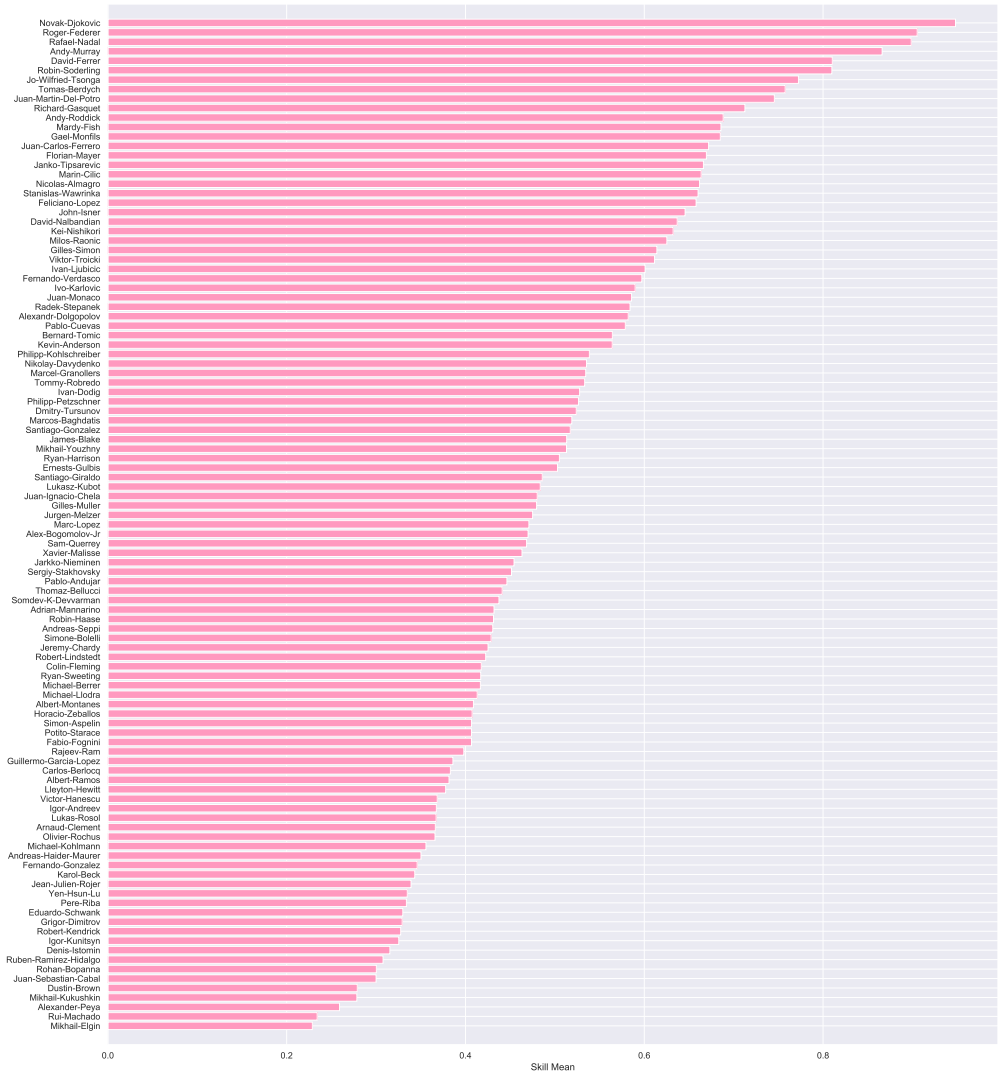


Figure 9: Ranking using game outcome averages - MP.

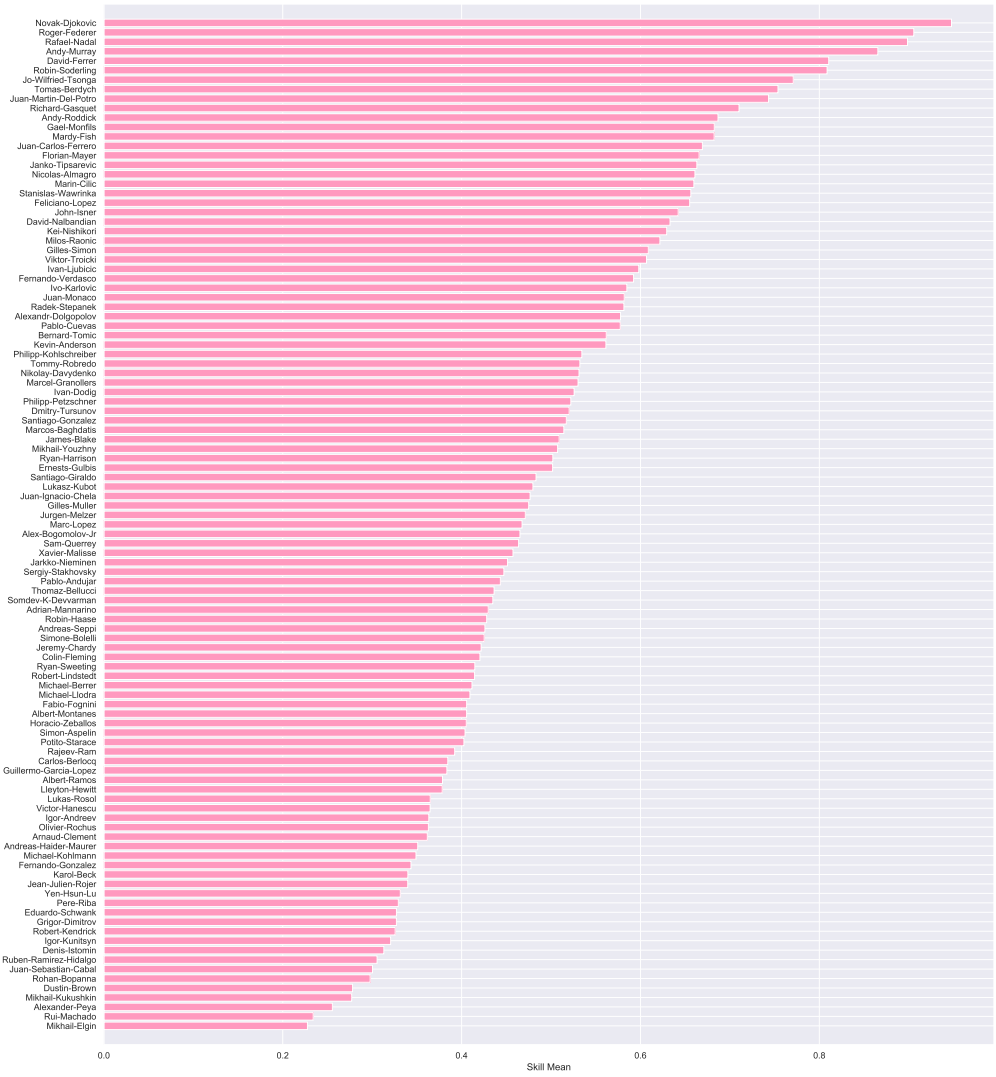


Figure 10: Ranking using game outcome averages - Gibbs.

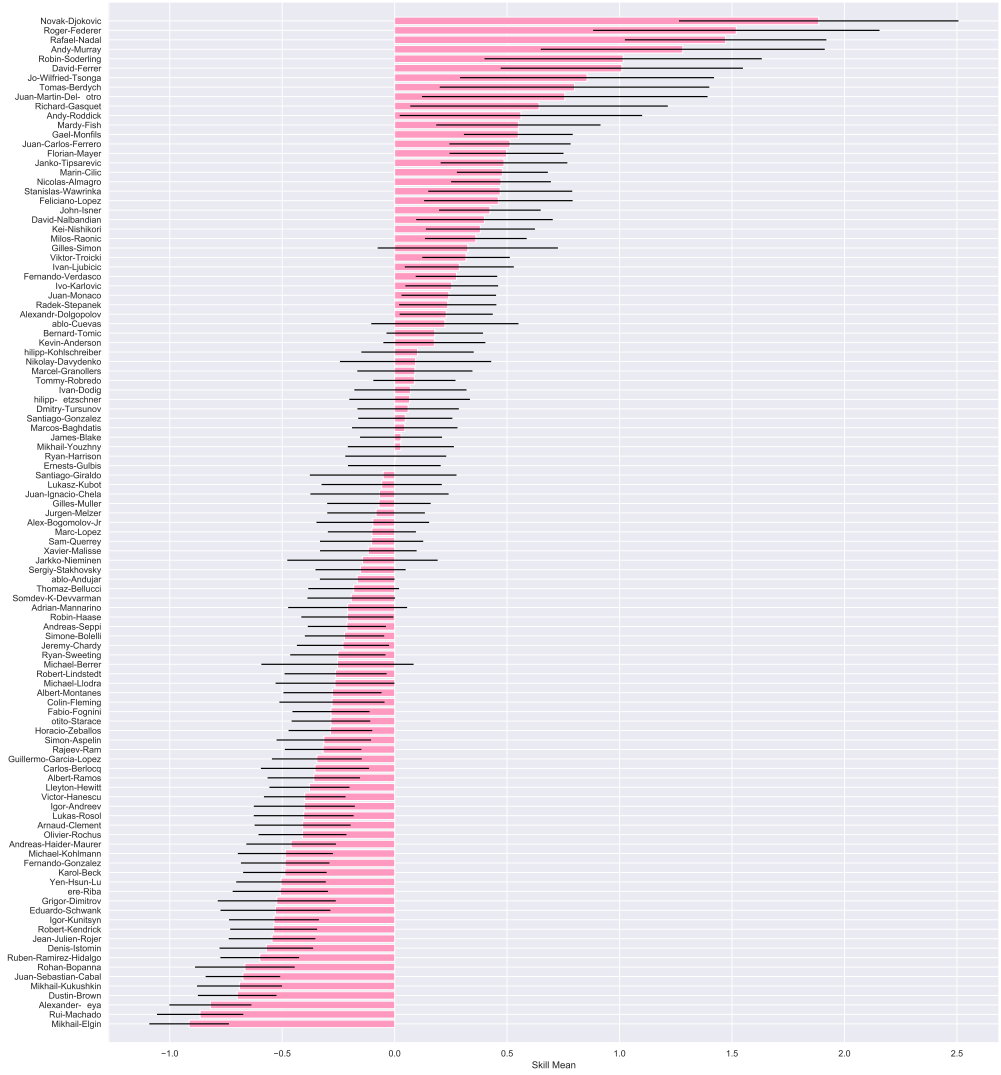


Figure 11: Ranking using expected skill - MP. Standard deviation is shown as black error bars.

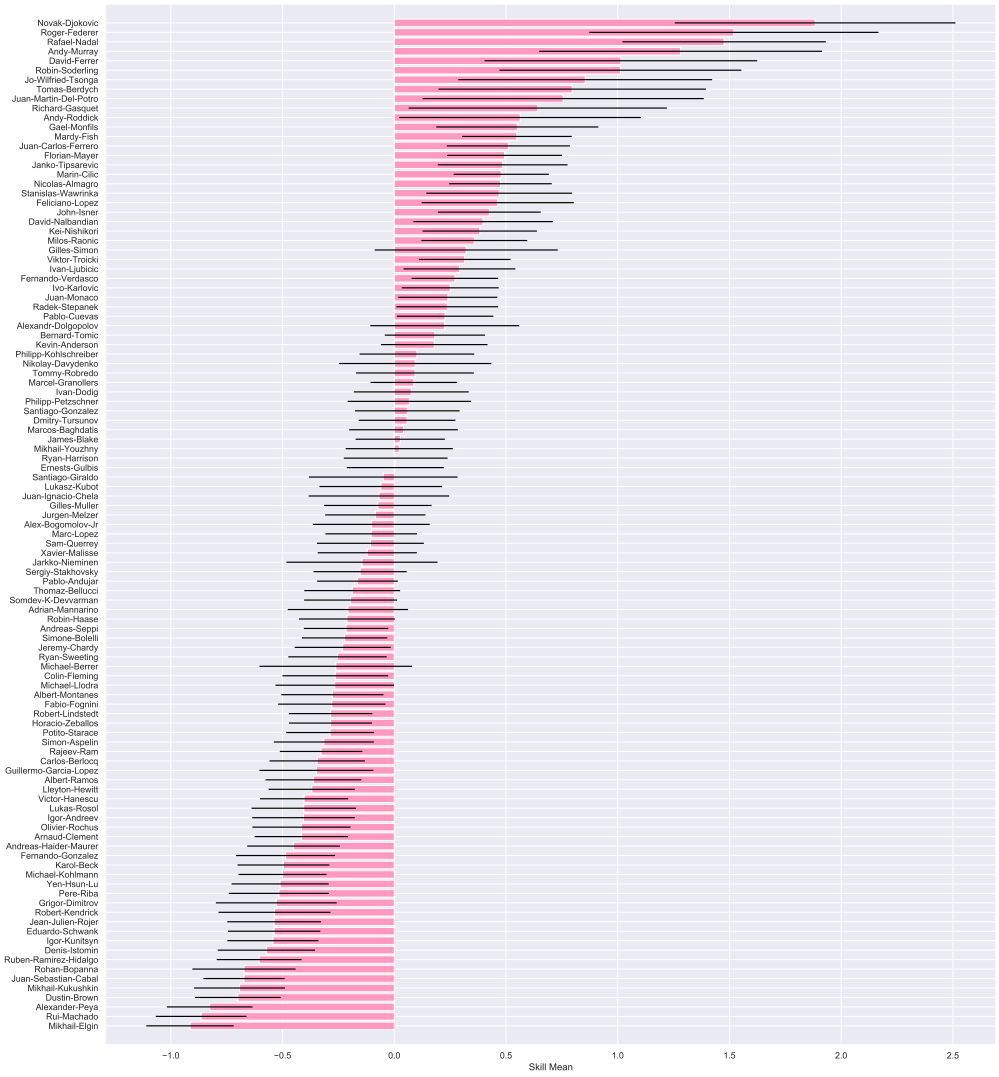


Figure 12: Ranking using expected skil - Gibbs. Standard deviation is shown as black error bars.