4F13 Probabilistic Machine Learning
Coursework 2: Probabilistic Ranking

Candidate Number: 5746G

November 20, 2020

Question a)

Listings 1: Conditional posterior mean over skills given performances.

```
outcomes = (G[:, 0] == p).astype(np.float32) - (G[:, 1] == p).astype(np.float32)
m[p] =  (outcomes * t[:, 0]).sum() # (prior mean's m_0[p] are zero for all players p)
```

Listings 2: Sum of precision matrices of conditional posterior over skills given performances.

```
iS[G[g, 0], G[g, 0]] += 1
iS[G[g, 1], G[g, 1]] += 1
iS[G[g, 0], G[g, 1]] -= 1
iS[G[g, 1], G[g, 0]] -= 1
```

Figure 1 plots sampled player skills as a function of Gibbs iteration for three different players. The early samples vary significantly, before settling on the *stationary distribution*, a region with constant mean and variance, which is explored fully by the sampler over the remaining iterations.

The *burn in* time refers to the number of iterations required for the Markov Chain of samples to reach the desired stationary distribution [3]. Since skill samples are generated *dependently* from the conditional posterior, initialising the Gibbs sampler (by means of random seed) far from the target distribution may result in over-sampling of regions that are assigned low probability under the equilibrium distribution. Using samples drawn from the posterior, the joint posterior distribution over the i-th sampled skills $\mathbf{w}^{(i)}$ given the observed game outcomes $\mathbf{y}$ takes the form:

$$
\begin{aligned}
p(\mathbf{w}^{(i)} \mid \mathbf{y}) &= \frac{p(\mathbf{w}^{(i)}, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{w}^{(i)})p(\mathbf{y} \mid \mathbf{w}^{(i)})}{p(\mathbf{y})} && (1)\\
&\propto \underbrace{p(\mathbf{w}^{(i)})}_{\text{prior}}\underbrace{p(\mathbf{y} \mid \mathbf{w}^{(i)})}_{\text{likelihood}} && (2)\\
&= p(\mathbf{w}^{(i)}) \prod_{g=1}^{G} p(y_g \mid w_{I_g}^{(i)}, w_{J_g}^{(i)}) && (3)\\
&= \mathcal{N}(\mathbf{w}^{(i)}; \mathbf{0}, \mathbf{I}) \prod_{g=1}^{G} \Phi\left(y_g(w_{I_g}^{(i)} - w_{J_g}^{(i)})\right) && (4)
\end{aligned}
$$

where the cumulative probability function $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(x; 0, 1)dx$. The logarithm of the joint probability $\log(p(\mathbf{w}^{(i)}, \mathbf{y}))$ is plot as a function of Gibbs iteration i in Figure 2. The probability of

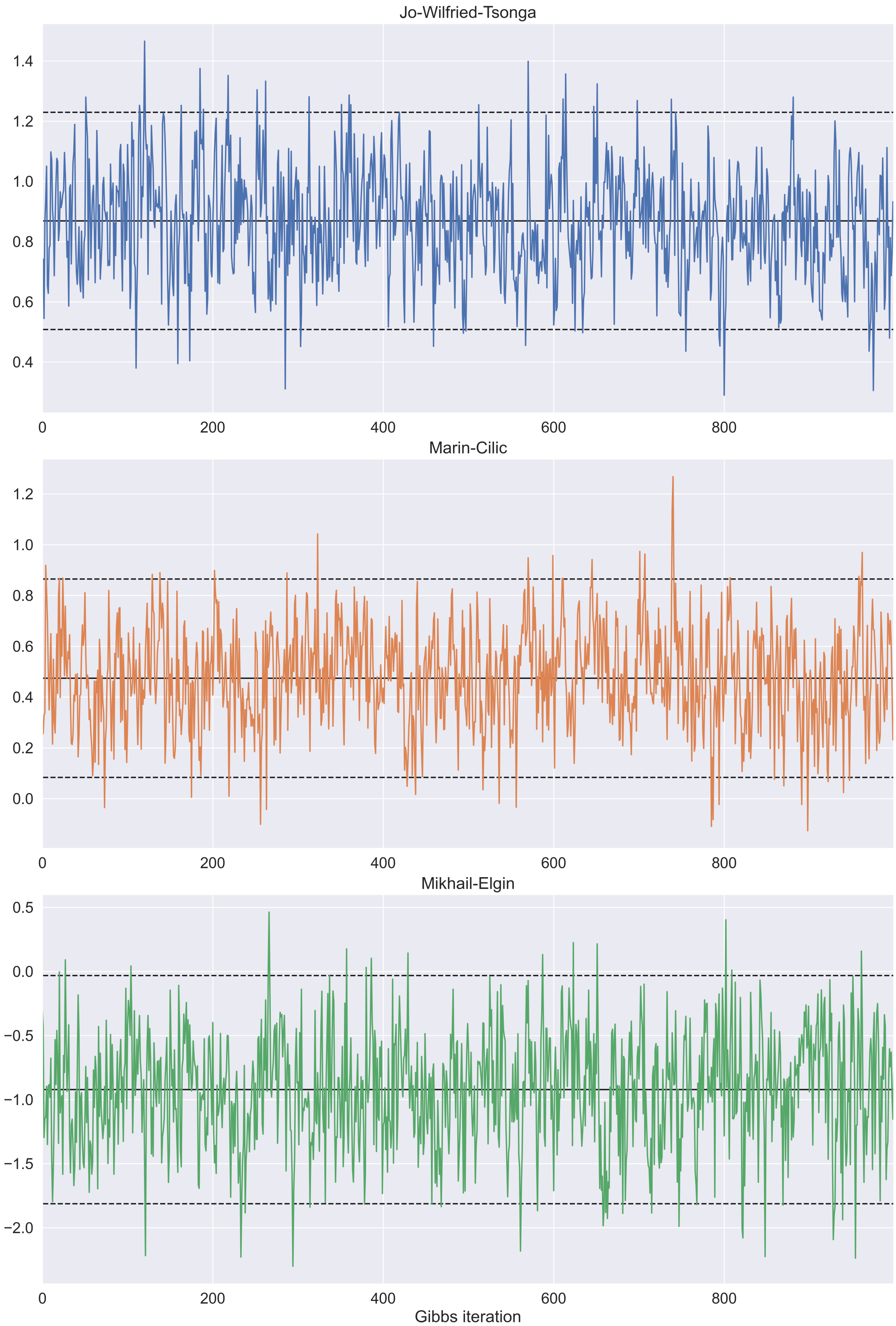


Figure 1: Sampled player skills as a function of Gibbs iteration, with steady state mean (solid) and plus and minus two times the standard deviation (dashed) shown in black.

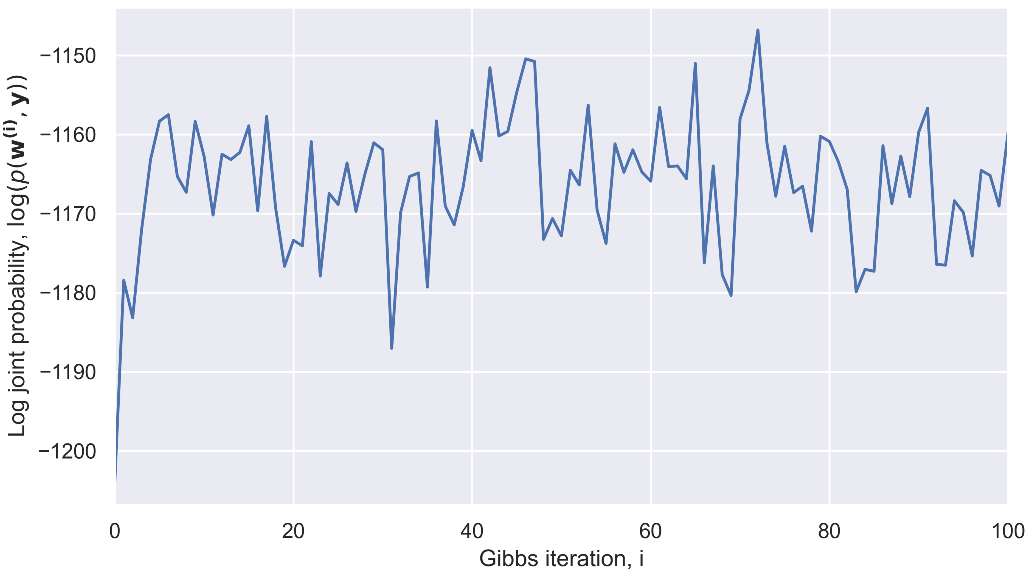


Figure 2: Logarithm of joint probability of player skills and game outcomes with Gibbs iterations.

drawing the first 5 or so samples is low, after which they are drawn from similarly high probability regions. This indicates a burn in period of ~ 5 samples. To account for the potentially longer mixing times which may occur for less favourable starting points, a conservative approach is taken, and the first 25 samples discarded.

From the Gibbs samples of skills drawn from the conditional posterior $p(\mathbf{w} \mid \mathbf{y})$, the Monte Carlo estimator for $\boldsymbol{\mu} = \mathbb{E}_{p(\mathbf{w}|\mathbf{y})}[\mathbf{w}]$ is given by $\hat{\boldsymbol{\mu}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{w}^{(i)}$. Defining the auto covariance of $\mathbf{w}$ at lag τ:

$$R_{\mathbf{w}}[\tau] = \mathrm{Cov}(\mathbf{w}^{(i)}, \mathbf{w}^{(i+\tau)}), \tag{5}$$

the auto correlation $\rho_{\mathbf{w}}[\tau]$:

$$\rho_{\mathbf{w}}[\tau] = \frac{R_{\mathbf{w}}[\tau]}{R_{\mathbf{w}}[0]}, \tag{6}$$

and assuming the process to be *weakly stationary*, the variance of this estimator is:

$$
\begin{aligned}
\mathbb{V}[\hat{\boldsymbol{\mu}}] &= \mathbb{V}\left[\frac{1}{T}\sum_{i=1}^{T}\mathbf{w}^{(i)}\right] && (7)\\
&= \frac{1}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T}\mathrm{Cov}(\mathbf{w}^{(i)},\mathbf{w}^{(j)}) && (8)\\
&= \frac{\mathbb{V}[\mathbf{w}]}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T}\frac{\mathrm{Cov}((\mathbf{w}^{(i)},f(\mathbf{w}^{(j)}))}{\mathbb{V}[\mathbf{w}^{(i)}]} && (9)\\
&= \frac{\mathbb{V}[\mathbf{w}]}{T^2}\sum_{i=1}^{T}\sum_{j=1}^{T}\frac{R_{\mathbf{w}}[i-j]}{R_{\mathbf{w}}[0]} && (10)\\
&= \frac{\mathbb{V}[\mathbf{w}]}{T^2}\left(T+2\sum_{i=1}^{T-1}(T-i)\frac{R_{\mathbf{w}}[i]}{R_{\mathbf{w}}[0]}\right) && (11)\\
&= \frac{\mathbb{V}[\mathbf{w}]}{T}\left(1+2\sum_{i=1}^{T-1}\left(1-\frac{i}{T}\right)\rho_{\mathbf{w}}[i]\right) && (12)\\
&\approx \frac{\mathbb{V}[\mathbf{w}]}{T}\left(1+2\sum_{i=1}^{T-1}\rho_{\mathbf{w}}[i]\right) \qquad \text{for sufficiently large } T && (13)\\
&= \frac{\mathbb{V}[\mathbf{w}]}{T_e} && (14)
\end{aligned}
$$

where the *effective sample size* T_e is the factor to which the variance of the skills is reduced by the sampler [2]:

$$T_e = \frac{T}{1+2\sum_{i=1}^{T-1}\rho_{\mathbf{w}}[i]}. \tag{15}$$

Figure 3 plots the auto covariance coefficients of Gibbs samples with time lag. For all players, the coefficients are negligible for time lag > 10. Hence, the effective sample size is approximated as:

$$T_e \approx \frac{T}{1+2\sum_{i=1}^{10}\rho_{\mathbf{w}}[i]}.$$

For 4000 Gibbs iterations, $T_e \approx \frac{4000}{1+2\sum_{i=1}^{10}\rho_{\mathbf{w}}[i]} = 1000$; this yields a significant reduction in the variance of sampled skills (a factor of 1000), indicating an appropriate sample size.

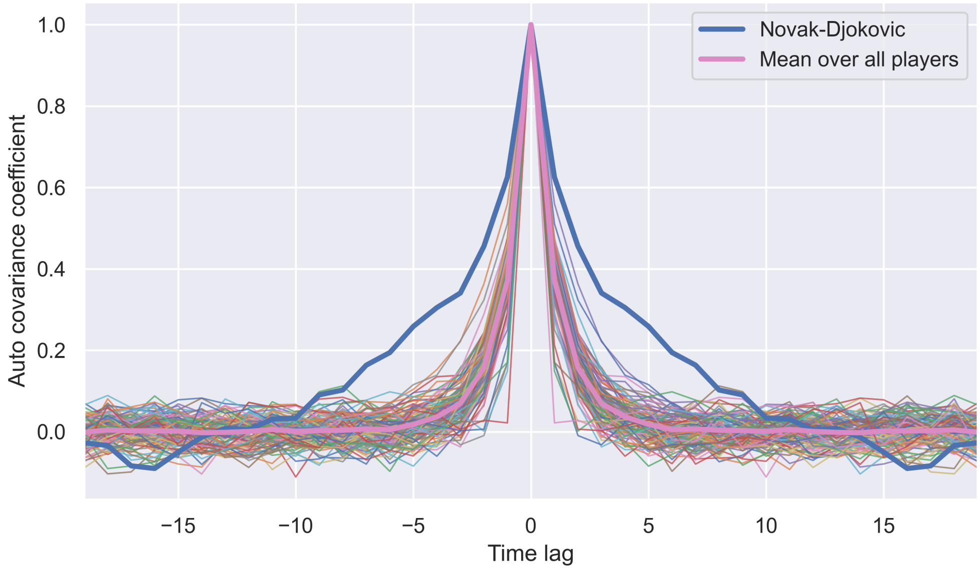


Figure 3: Auto covariance coefficient of Gibbs samples with lag. The player with the slowest decaying coefficients (Novak Djokovic) is highlighted.

Question b)

In the case of the Gibbs sampler, convergence is achieved when the samples are drawn from the stationary (or invariant) distribution of the transition kernel which defines the Markov Chain. As discussed in a), this is the case when the properties of the sampled distribution (mean and variance) remain steady over many iterations. From Figure 2, this was shown to be the case after ~ 5 iterations.

Since the TrueSkill graph is not a tree, the message passing algorithm is an *iterative* process [1]. Consequently, convergence can be defined as the point at which the absolute difference between successive estimates for the marginal means and precisions for all players fall within prescribed intervals $\epsilon_\mu, \epsilon_\lambda > 0$:

$$|\mu_j^{(i+1)} - \mu_j^{(i)}| < \epsilon_\mu \quad \text{and} \quad |\lambda_j^{(i+1)} - \lambda_j^{(i)}| < \epsilon_\lambda \quad \text{for } j = 1, \ldots, M \tag{16}$$

Figure 4 plots the maximum absolute change between successive marginal mean and precisions over all players for the message passing algorithm. Taking $\epsilon_\mu = \epsilon_\lambda = \epsilon = 10^{-4}$, the algorithm can be said to have converged after 55 iterations.

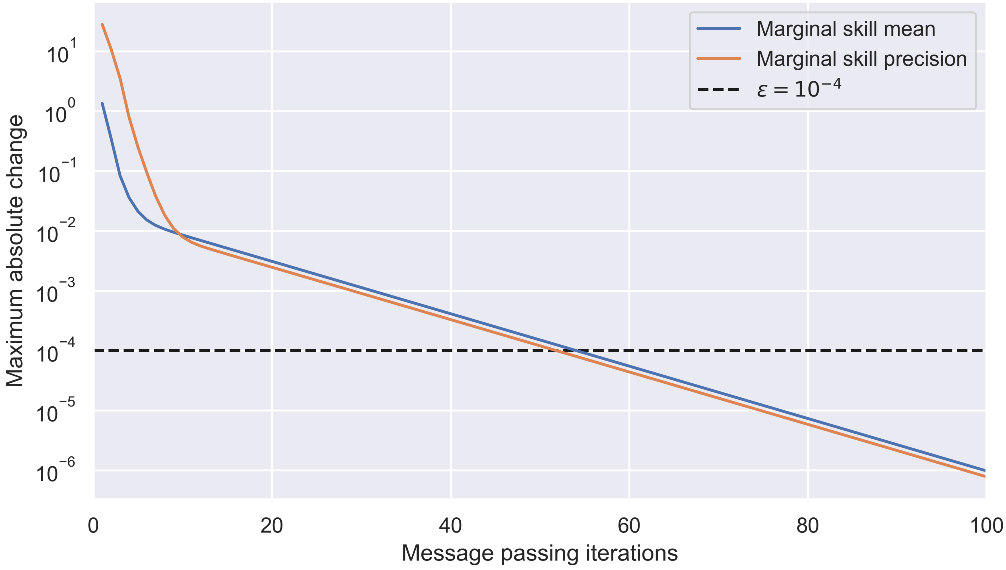


Figure 4: Maximum absolute change of marginal mean and precisions for player skills between successive message passing iterations.

Question c)

Listings 3: Probability that the skill of Player i is greater than that of Player j.

```
p_skill[i,j] = scipy.stats.norm.cdf(0, mu[j] - mu[i], np.sqrt(var[i] -
    var[j]))
```

Listings 4: Probability that Player i wins against Player j.

```
p_skill[i,j] = scipy.stats.norm.cdf(0, mu[j] - mu[i], np.sqrt(var[i] -
    var[j] + 1))
```

The message passing algorithm returns the posterior marginal distribution of player skills as independent Gaussian's of the form:

$$p(w_j) = \mathcal{N}(w_j; \mu_j, \sigma_j^2) \tag{17}$$

Let $z = w_2 - w_1$, such that:

$$p(z) = \mathcal{N}(z; \mu_2 - \mu_1, \sigma_1^2 + \sigma_2^2). \tag{18}$$

Table 1: Probability that the skill of Player 1 is greater than that of Player 2, computed using 100 iterations of the message passing algorithm.

		Player 2			
		Novak Djokovic	Rafael Nadal	Roger Federer	Andy Murray
Player 1	Novak Djokovic	-	0.9398	0.9089	0.9853
	Rafael Nadal	0.0602	-	0.4272	0.7665
	Roger Federer	0.0911	0.5728	-	0.8108
	Andy Murray	0.0147	0.2335	0.1892	-

Table 2: Probability that Player 1 wins against Player 2, computed using 100 iterations of the message passing algorithm.

		Player 2			
		Novak Djokovic	Rafael Nadal	Roger Federer	Andy Murray
Player 1	Novak Djokovic	-	0.6554	0.6380	0.7198
	Rafael Nadal	0.3446	-	0.4816	0.5731
	Roger Federer	0.3620	0.5184	-	0.5909
	Andy Murray	0.2802	0.4269	0.4091	-

The probability that Player 1 has a higher skill than Player 2 is then:

$$
\begin{aligned}
p(w_1 > w_2) &= p(w_2 - w_1 < 0) = p(z < 0) && (19)\\
&= \int_{-\infty}^{0} \mathcal{N}(z; \mu_2 - \mu_1, \sigma_1^2 + \sigma_2^2) dz && (20)\\
&= \Phi\left(\frac{\mu_2 - \mu_1}{\sigma_1^2 + \sigma_2^2}\right). && (21)
\end{aligned}
$$

The game outcome is determined by the sign of the *performance difference* t:

$$t = w_1 - w_2 + \eta \tag{22}$$

where $\eta \sim \mathcal{N}(0, 1)$, such that:

$$p(t) = \mathcal{N}(t; \mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2 + 1). \tag{23}$$

Player 1 is said to have won the game when $t > 0$:

$$p(t > 0) = \Phi\left(\frac{\mu_2 - \mu_1}{\sigma_1^2 + \sigma_2^2 + 1}\right). \tag{24}$$

In comparing the skill and win probabilities in Tables 1 and 2, it can be seen that the player predicted of having higher skill receives greater probability of winning. However, the confidence in predicting game outcomes (wins) is lesser than the corresponding skill prediction; the added noise term η inflates the variance of the win predictions, resulting in win probabilities being much closer to 0.5.

Question d)

Listings 5: Probability that the skill of Player i is greater than that of Player j, approximating their joint skills as a Gaussian.

```
joint_gaussian = scipy.stats.multivariate_normal(mean=mean_vector,
    cov=cov_matrix)
f = lambda w1, w2: joint_gaussian.pdf([w1, w2])
p_joint, abserr = scipy.integrate.dblquad(f, -5, 5, lambda x: x, lambda x:
    5)
```

Table 3 compares the skills of Novak Djokovic and Rafael Nadal using the Gibbs samples in three different ways: two of these methods involve approximating their skills by Gaussians, in which the Gaussian parameters are set to be those calculated empirically from the generated samples, whilst the third uses the samples directly.

The prediction for Gaussian marginal skills is given in eq. (21), and is obtained through use of Listings 3. It assumes that the player skills are uncorrelated. In reality, this is not the case since

Table 3: Probability that the skill of Novak Djokovic (N.D.) is greater than that of Rafael Nadal (R.N.), calculated using three methods based on Gibbs samples.

	Marginal Gaussians	Joint Gaussian	Direct Estimate
p($w_{N.D.} > w_{R.N.}$)	0.910	0.941	0.950

Table 4: Probability that the skill of Player 1 is greater than that of Player 2, computed using Gibbs samples directly.

		Player 2			
		Novak Djokovic	Rafael Nadal	Roger Federer	Andy Murray
Player 1	Novak Djokovic	-	0.9517	0.9008	0.9875
	Rafael Nadal	0.0483	-	0.4442	0.7908
	Roger Federer	0.0992	0.5558	-	0.8133
	Andy Murray	0.0125	0.2092	0.1867	-

the Gibbs samples are drawn from the joint distribution of player skills given the game outcomes $p(\mathbf{w} \mid \mathbf{y})$.

The joint skill approximation accounts for said correlation, modelling the skills as a multivariate Gaussian distribution:

$$p\left(w_1, w_2\right) = N\left(\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}; \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho\left(w_1, w_2\right) \\ \rho\left(w_1, w_2\right) & \sigma_2^2 \end{bmatrix}\right), \tag{25}$$

and as such is a better approximation than the marginal method. The required probability is then:

$$p\left(w_1 > w_2\right) = \int_{-\infty}^{\infty} \int_{-\infty}^{w_1} p\left(w_1, w_2\right) dw_2 dw_1, \tag{26}$$

which is found through use of the SciPy library functions `multivariate_normal`[1] and `integrate.dblquad`[2], as outlined in Listings 5.

The third and final approach uses the Gibbs samples directly by calculating the proportion of samples for which w_1 is greater than w_2. It makes no assumption about the distribution of skills, and is therefore likely to be the best method for comparing player skills, since the true distribution over skills is not Gaussian.

Table 4 shows the probabilities that the skill of one player is higher than the other calculated directly from the Gibbs samples. These probabilities are in close agreement to those returned from the message passing algorithm, shown in Table 1, with a mean absolute difference of 0.015.

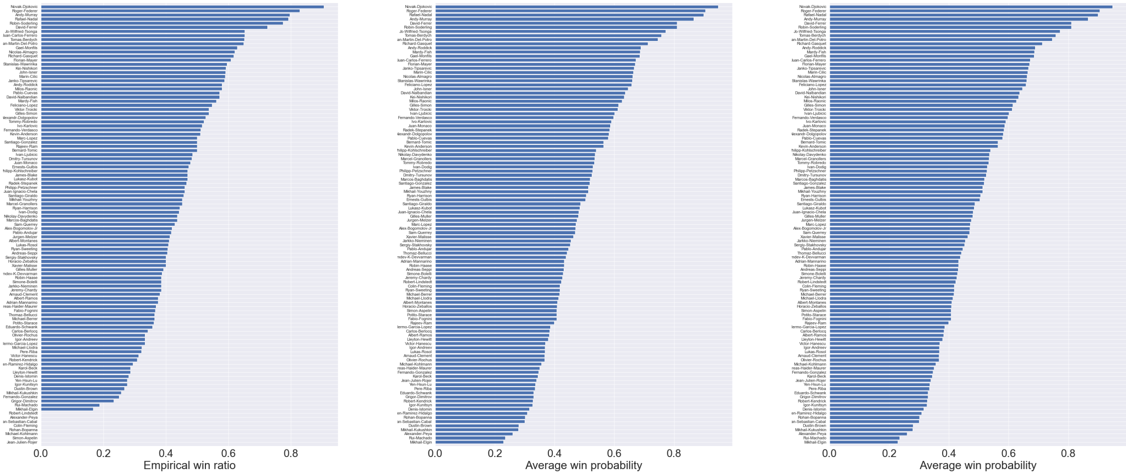
Question e)

Figure 5 plots the player rankings using predicted outcomes for three different methods of inference. The x-axis values can be interpreted as the probability of a given player winning against any other player chosen at random.

Figure 5(a) shows simply the empirical win ratio. This ranking system takes no account of the skill of the opponent. As such, players who win the same fraction of games will receive the same ranking. Furthermore, it is highly unstable: any player who plays a single game and wins will receive

[1]SciPy `multivariate_normal`: `https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.stats.multivariate_normal.html`

[2]SciPy `integrate.dblquad`: `https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.dblquad.html`



(a) Empirical game outcome averages.

(b) Predictions based on Gibbs samples.

(c) Predictions based on the message passing algorithm.

Figure 5: Player rankings using predicted outcomes for three different methods of inference. All are shown full size in the Appendix.

the highest possible ranking (with an average win ratio of 1), whereas any player who does not win a match will be ranked joint last (with an average win ratio of 0).

Probabilistic approaches overcome these issues by calculating each player's expected probability of winning. For given skill means and variances, these probabilities are calculated through use of Listings 4. Figure 5(b) shows the rankings obtained using the Gibbs samples directly, as described in Question (d), whereas Figure 5(c) is generated from the marginal means and variances returned by the message passing algorithm. Both of these frameworks return rankings for which opponent skill is taken into account: for example, Robert Lindstedt is ranked higher than Mikhail Elgin, despite the former winning no games and the latter winning near 20%.

Word Count (excluding Listings): 1000

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[2] Jonathan Goodman and Jonathan Weare. Ensemble samplers with affine invariance. *Communications in Applied Mathematics and Computational Science*, 5, 2010. doi: 10.2140/camcos.2010.5.65.

[3] David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. ISBN 9780521642989.

Appendix

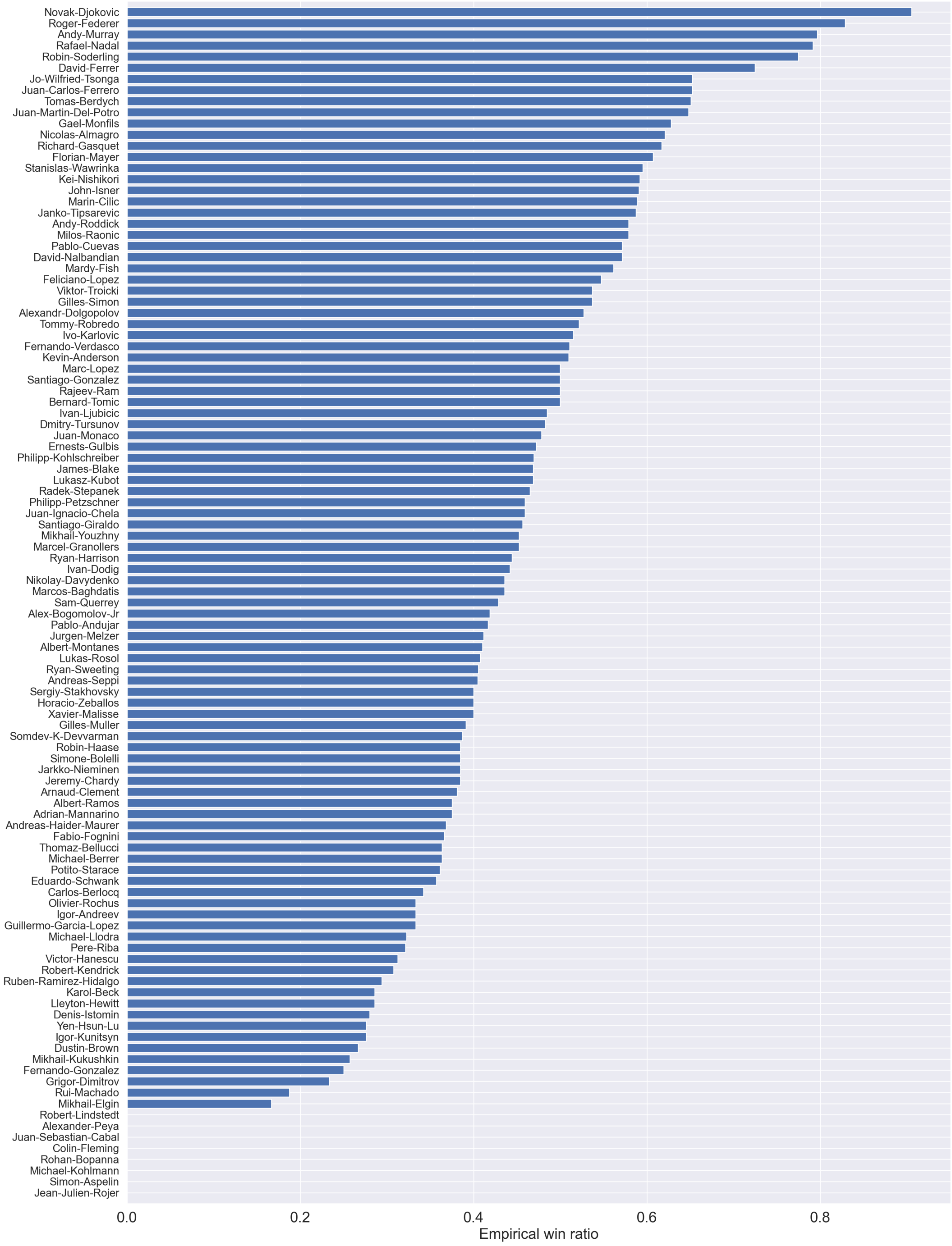


Figure 6: Player rankings using empirical game outcome averages.

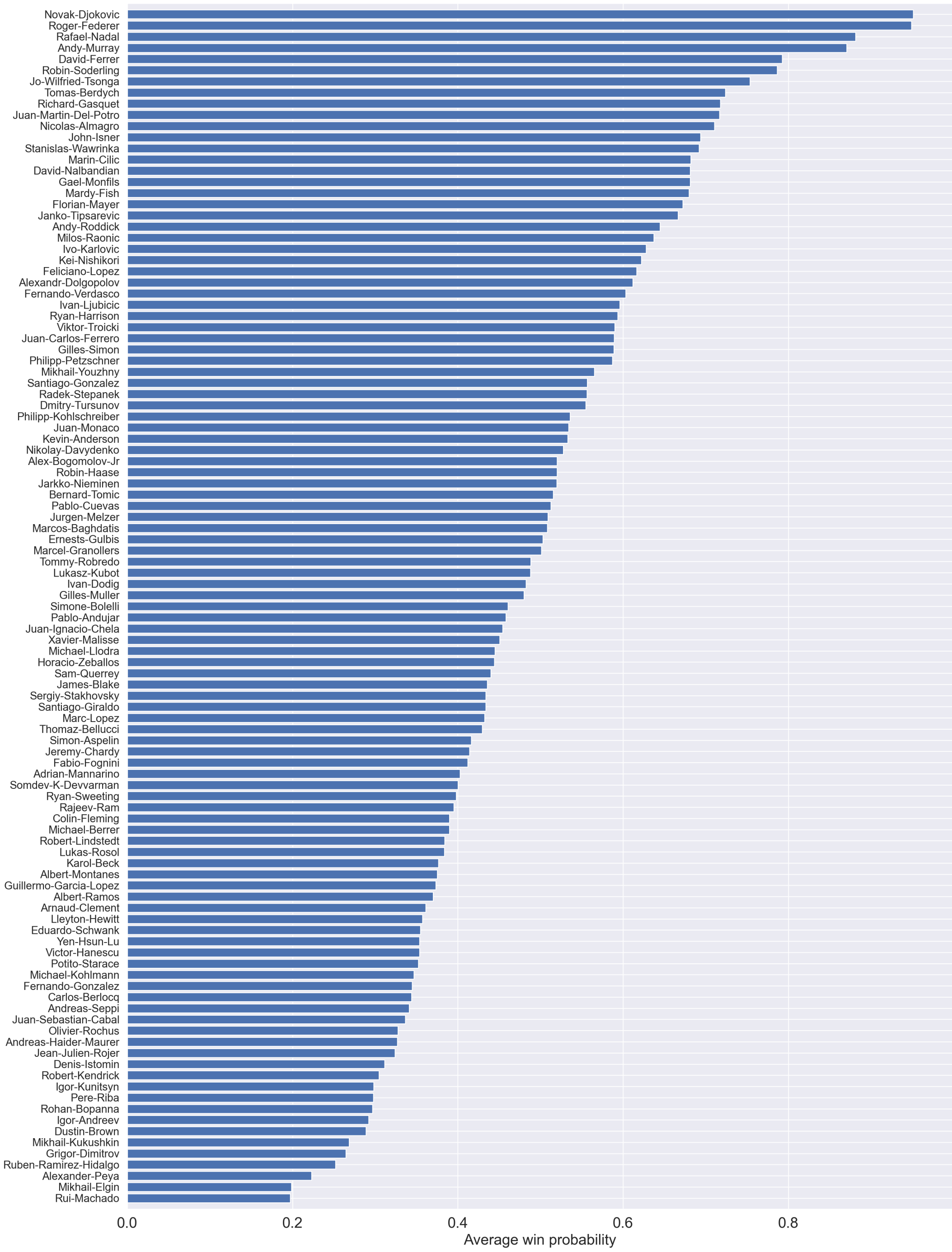


Figure 7: Player rankings using predictions based on Gibbs samples

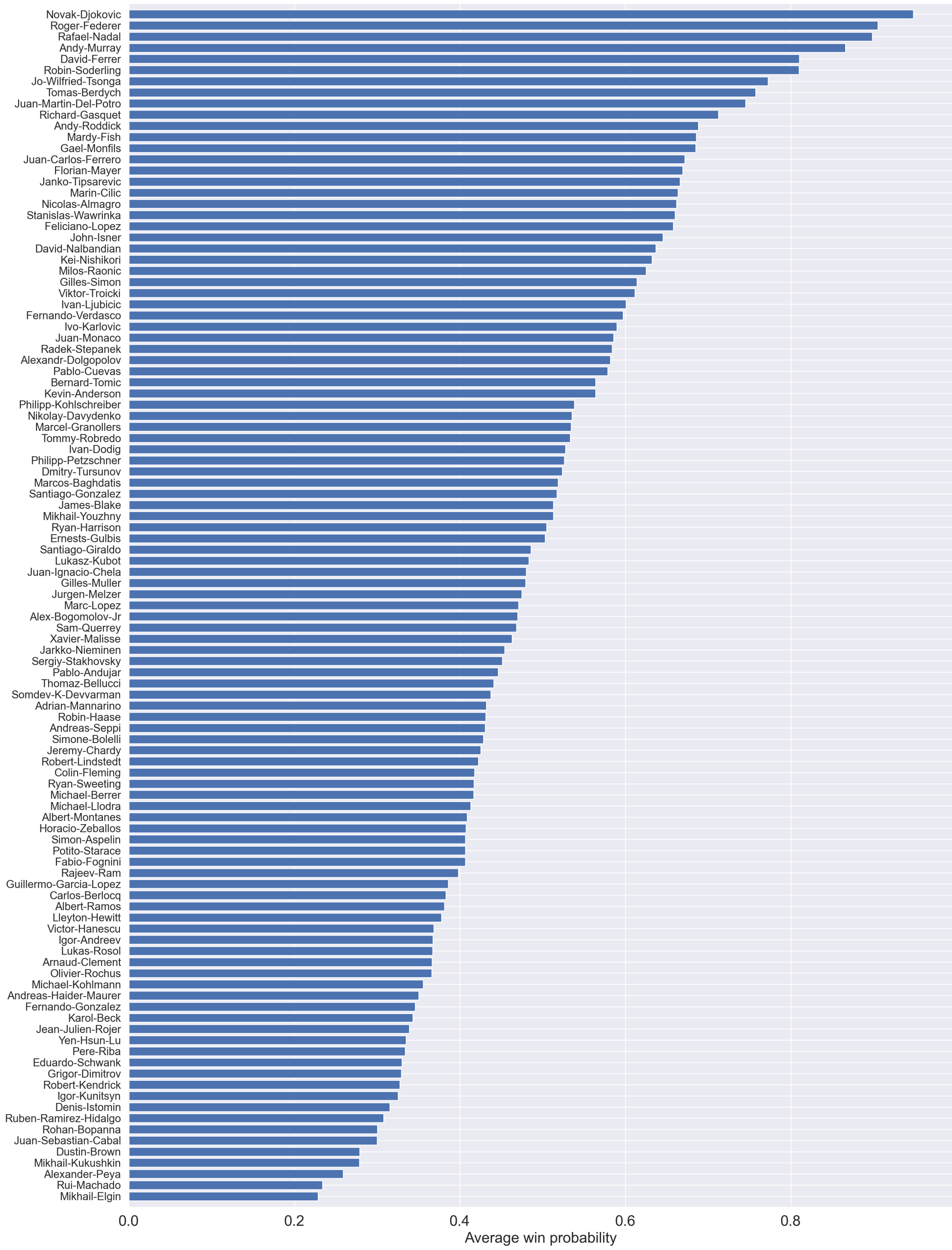


Figure 8: Player rankings using predictions based on the message passing algorithm.