CUED - Engineering Tripos Part IIB 2021-2022 **Module Coursework**

Module	4F13	Title of report	Coursework 2: Probabilistic Ranking

Date submitted: 18/11/2022	Assessment for this module is ☑ 100% / □ 25% coursework of which this assignment forms 33 %

UNDERGRADUATE STUDENTS ONLY		**POST GRADUATE STUDENTS ONLY**			
Candidate number:	5554D	Name:		College:	

Feedback to the student

□ **See also comments in the text**

		Very good	**Good**	Needs improvmt
C O N T E N T	**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:			
P R E S E N T A T I O N	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?			
	Comments:			

Indicative grades are not provided for the FINAL piece of coursework in a module

Assessment (circle one or two grades)		A*	A	B	C	D
Indicative grade guideline		>75%	65-75%	55-65%	40-55%	<40%
Penalty for lateness:		*20% of maximum achievable marks per week or part week that the work is late.*				

Marker: Date:

4F13 Probabilistic Machine Learning
Coursework 2: Probabilistic Ranking

CCN: 5554D
Words: 999

November 18, 2022

1 Task A

Gibbs sampling is a Markov chain Monte Carlo (MCMC) method for sequentially sampling from a multivariate distribution. The Gibbs sampler was run for 1100 iterations, and the sampled skills of four players against Gibbs iteration are shown in Figure 1. Initial samples are not representative of the equilibrium distribution, instead over-sampling regions of low probability - some time is needed for the Markov chain to stabilise to its invariant distribution. Figure 2 shows the running averages of the skills of 7 players, and suggests that convergence occurs after 100 iterations - burn-in time will be set to 120, as a safety measure. A major disadvantage of this method is strong dependencies between consecutive samples - hence, samples are thinned by selecting every n$^{\text{th}}$ sample. An autocorrelation of about zero is required - therefore, from Figure 3, the autocorrelation time was set to $n = 10$. To obtain reliable results, a decent number of samples are required, thus the Gibbs sampler will be run for 2120 iterations from now on, providing 200 samples.

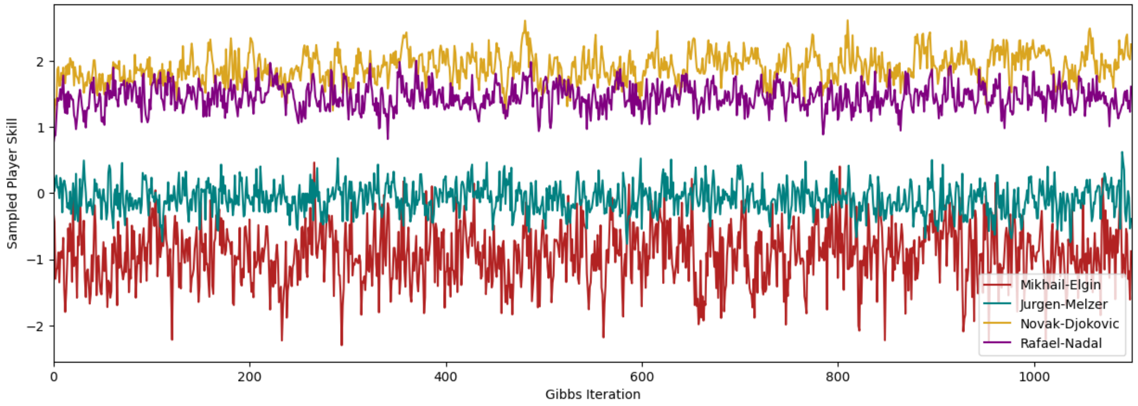

Figure 1: Sampled player skill against Gibbs iteration, for 1100 iterations.

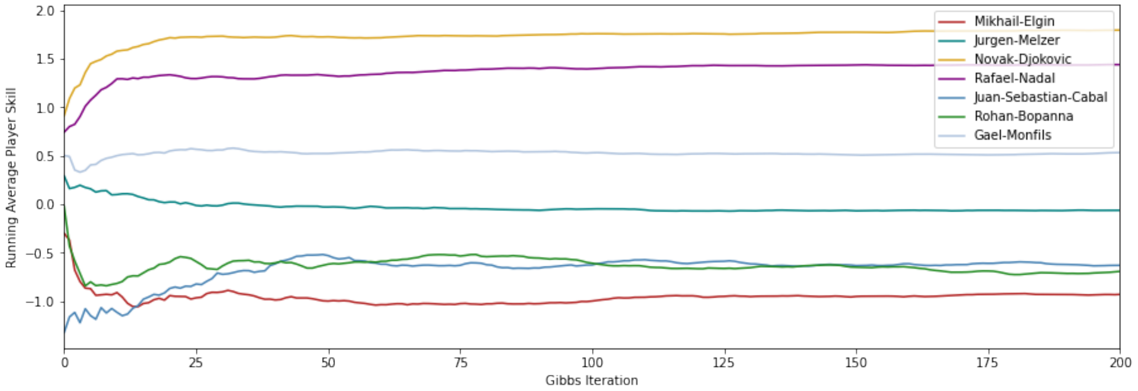

Figure 2: Running average of player skill from 1st iteration to n$^{\text{th}}$ iteration, as a function of the Gibbs iteration, for 200 iterations.

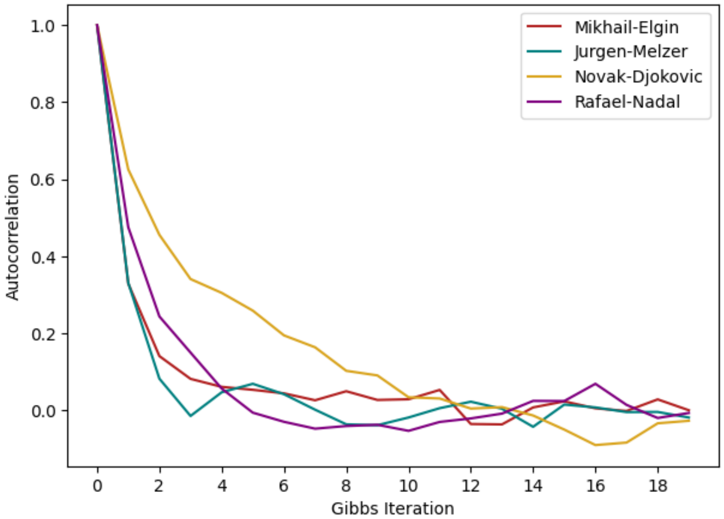

Figure 3: Autocorrelation against Gibbs iteration

Listing 1: Code snippet to sample from the conditional distributions needed for Gibbs sampling.

```
# Jointly sample skills given performance differences
m = np.zeros((M, 1))
for p in range(M):
    wins = np.array(G[:, 0] == p).astype(int)
    losses = np.array(G[:, 1] == p).astype(int)
    m[p] = np.dot(t.flatten(), wins-losses)

iS = np.zeros((M, M)) # Container for sum of precision matrices (likelihood terms)

for g in range(N):
    I_g, J_g = G[g]

    iS[I_g, I_g] += 1
    iS[I_g, J_g] -= 1
    iS[J_g, I_g] -= 1
    iS[J_g, J_g] += 1
```

2 Task B

In Gibbs sampling, the joint multivariate distribution converges to the true joint skill posterior. As mentioned in the previous section, convergence seems to take 100 iterations. To formally confirm that the thinned and burned-in samples have converged, the Geweke [1] diagnostic was used to compare the statistics of the first 10% and last 50% of samples. For each player, a Geweke z-score is calculated:

$$\text{score} = \frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}} \tag{1}$$

where μ_1 and μ_2 are the means and σ_1 and σ_2 are the standard deviations of the first and last sets, respectively. Figure 4 shows the scores for each player, which are all below 0.5, and therefore the Markov chain has successfully converged.

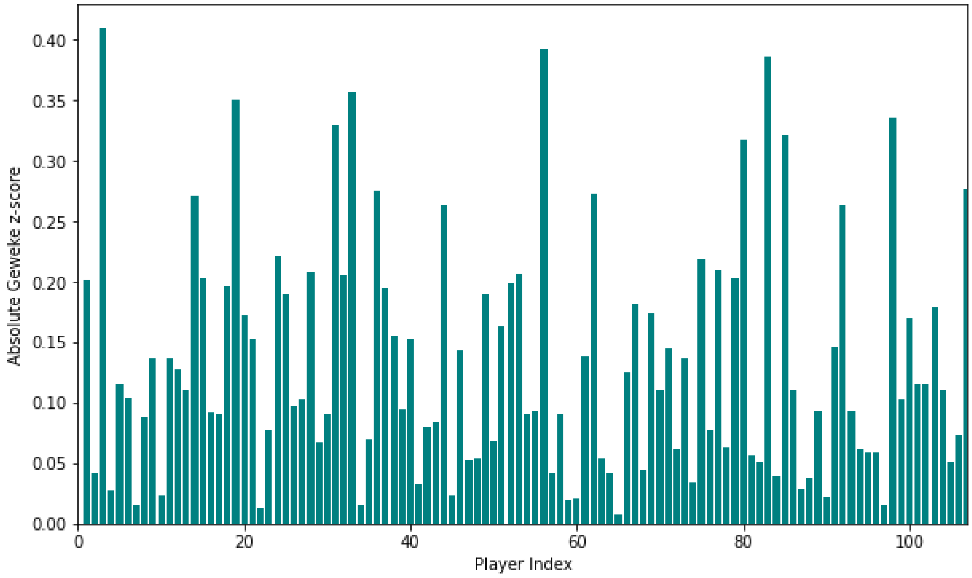

Figure 4: Absolute Geweke z-scores for each player.

Another method is message passing (MP) and expectation propagation (EP), which converges to the marginal skill distribution for each player, approximated by Gaussians, rather than the true joint distribution in Gibbs sampling. The player skill means and precisions (inverse variance) over 20 iterations are shown in Figure 5

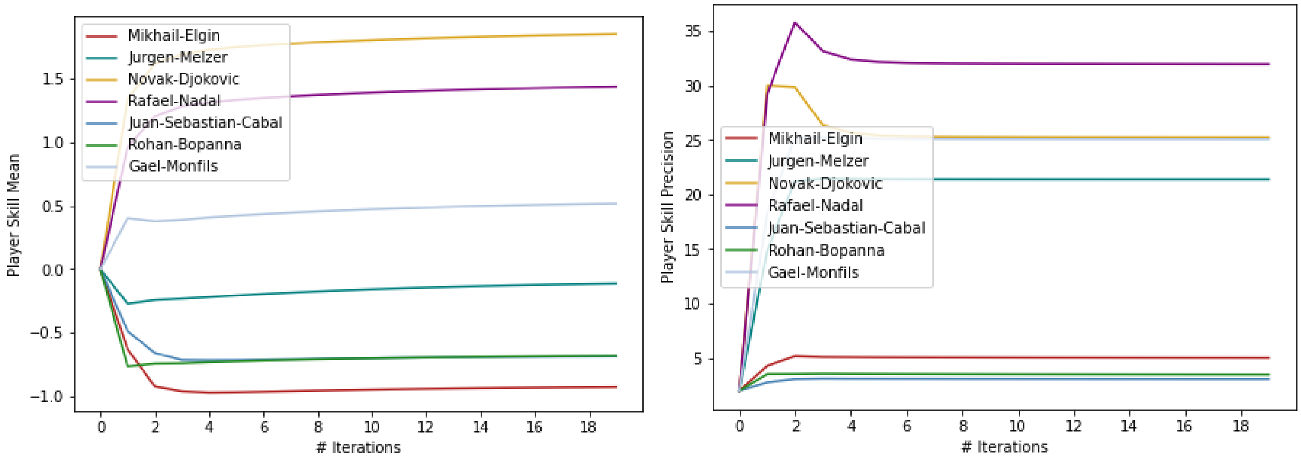

Figure 5: Player skill means and precisions over 20 iterations of message passing for the same 4 players.

The absolute changes in the means and precisions after each iteration are shown in Figure 6, which fall below 0.01 within 10 iterations, at which point the algorithm has converged - much quicker than Gibbs sampling.

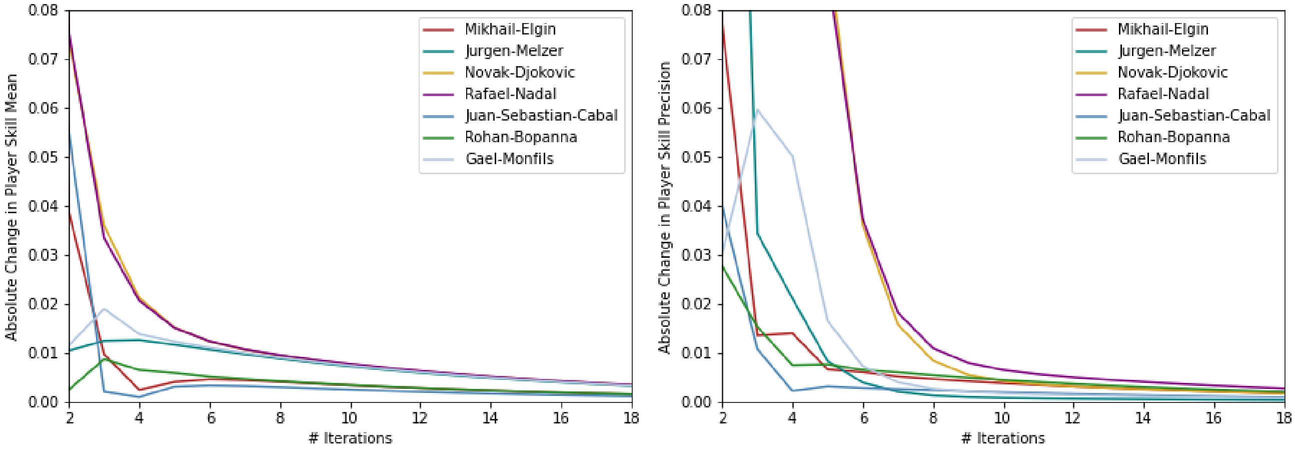

Figure 6: Absolute changes in layer skill means and precisions.

Listing 2: Code to burn-in and thin samples, and calculate the Geweke z-scores for each player.

```
#Burn-in and thin
burn_in = 120
thinning = 10
idxs = range(burn_in, num_iters, thinning)
thinned_samples = skill_samples[:, idxs]

#Geweke diagnostic
i1 = int(0.1 * len(thinned_samples[0]))
i2 = int(0.5 * len(thinned_samples[0]))

seq1 = thinned_samples[:, :i1] #first 10% for all players
seq2 = thinned_samples[:, i2:] #last 50% for all players

mu1, mu2 = np.mean(seq1, axis=1), np.mean(seq2, axis=1)
var1, var2 = np.var(seq1, axis=1), np.var(seq2, axis=1)
geweke_z_score = (mu1 - mu2) / np.sqrt(var1 + var2)
```

3 Task C

For the four highest rated players according to the ATP rankings in December 2011, the probabilities of a player having greater skill than another and a player winning a match were calculated from the player skill means and precisions output by the EP algorithm, and are provided in Table 1.

The marginal skill of player i is approximately Gaussian and denoted $w_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. The probability of a player being more skillful than another is given by

$$p(w_i > w_j) = p(w_i - w_j > 0) = 1 - \Phi\left(\frac{\mu_j - \mu_i}{\sqrt{\sigma_i^2 + \sigma_j^2}}\right) \tag{2}$$

Due to uncertainties in matches, noise is added to the performance difference: $n \sim \mathcal{N}(0, 1)$, and therefore the probability of a player winning a match is given by

$$p(w_i - w_j + n > 0) = 1 - \Phi\left(\frac{\mu_j - \mu_i}{\sqrt{\sigma_i^2 + \sigma_j^2 + 1}}\right) \tag{3}$$

As can be seen in Table 1, the probabilities of winning a match are less extreme than the probabilities of being more skillful, with probabilities closer to 0.5. Being more skillful does not necessarily mean that a player will win a given match - there are inconsistencies between player performances. A more skillful player will win more matches but not necessarily every match.

	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.940	0.909	0.985
Nadal	0.060	-	0.427	0.766
Federer	0.091	0.573	-	0.811
Murray	0.015	0.234	0.189	-

	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.655	0.638	0.720
Nadal	0.345	-	0.482	0.573
Federer	0.362	0.518	-	0.591
Murray	0.280	0.427	0.409	-

Table 1: Left: Probability that row player has greater skill than column player.
Right: Probability that row player beats column player in a game.

Listing 3: Code for calulating probabilities of a player having greater skill and winning a match from EP means and precisions.

```
for i in range(4):
    for j in range(4):
        if i == j:
            continue
        m1 = mean_player_skills[top_four[i]]
        m2 = mean_player_skills[top_four[j]]
        p1 = precision_player_skills[top_four[i]]
        p2 = precision_player_skills[top_four[j]]
        m = m2 - m1
        var = 1/p1 + 1/p2
        skills_probs[i,j] = 1 - norm.cdf(m/var**0.5)
        wins_probs[i,j] = 1 - norm.cdf(m/(var+1)**0.5)
```

4 Task D

Using the Gibbs sampler, the skills of Nadal and Djokovic can be compared by three different methods:

1. Approximating marginal skills as Gaussians,
2. Approximating joint skill as a multivariate Gaussian,
3. Empirically comparing joint samples:
    (a) 2120 iterations (200 samples),
    (b) 5120 iterations (500 samples)

The results are given in Table 2.

Method	1.	2.	3.a)	3.b)
p(Djokovic >Nadal)	0.937	0.959	0.965	0.946

Table 2: Probability of Djokovic being more skillful than Nadal, computed by 3 different methods.

Figure 7 shows the marginals for the two players - Djokovic has a greater mean and variance.

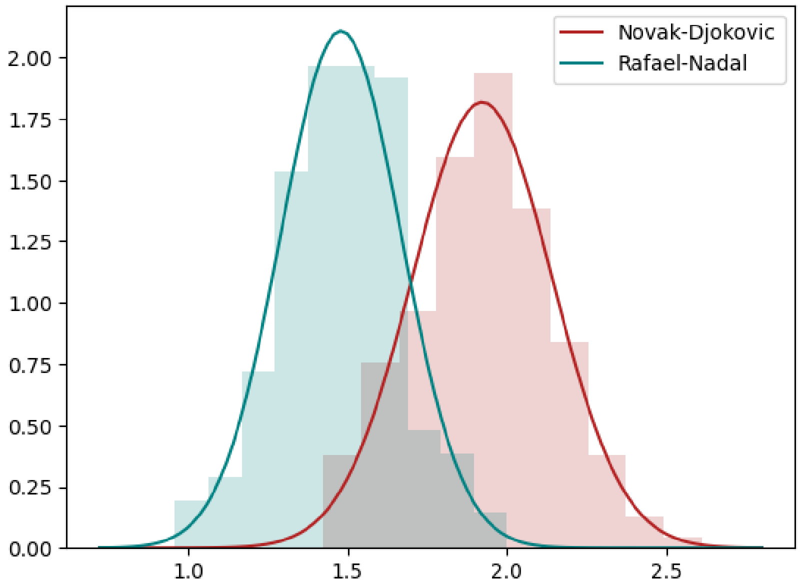

Figure 7: Approximate Gaussian marginals from Gibbs samples for Djokovic and Nadal.

Figure 8 shows the approximate joint Gaussian. Method 2 considers the covariance (which is non-zero), and is hence more accurate than method 1. However, method 2 is slower, as a large number of random variables must be sampled from the multivariate distribution, as the integral is not tractable.

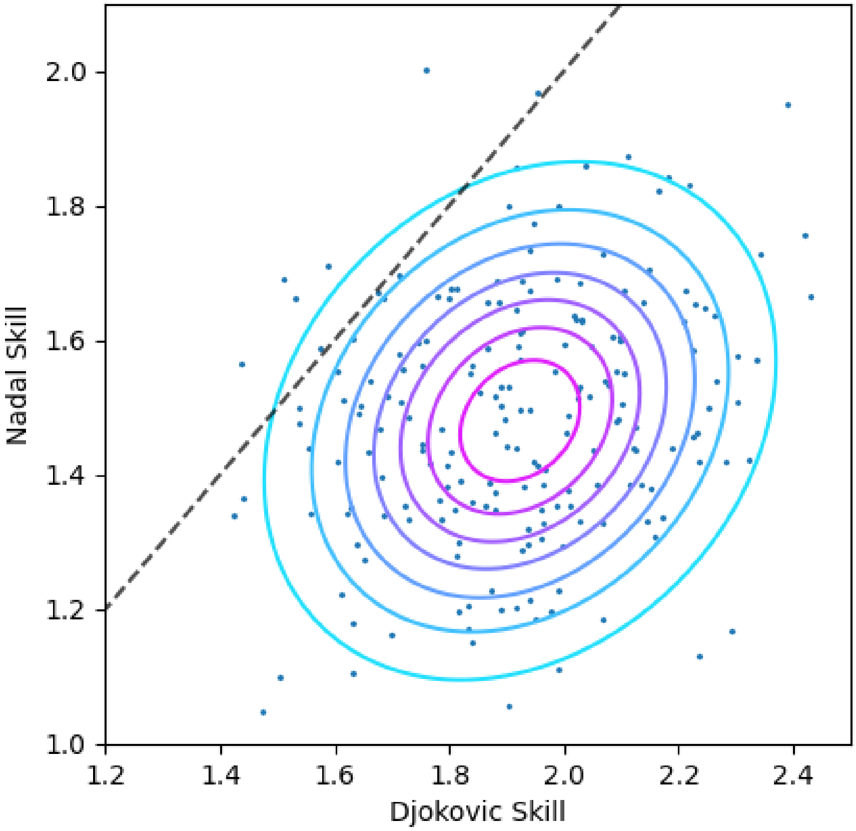

Figure 8: Contour plot of the approximated Gaussian joint distribution for Djokovic and Nadal.

The third method does not make any assumptions regarding the distribution of the data, which may not necessarily be Gaussian. Assuming that the Gibbs sampler correctly converges to the true joint posterior, the third method should be the most accurate. However, a large number of samples is required for this method to be accurate; Figure 9 shows how the empirical probabilities vary with the number of samples. The probability seems to stabilise after 400 samples, which requires

4120 Gibbs iterations due to burn-in and thinning. The previous two methods require fewer samples.

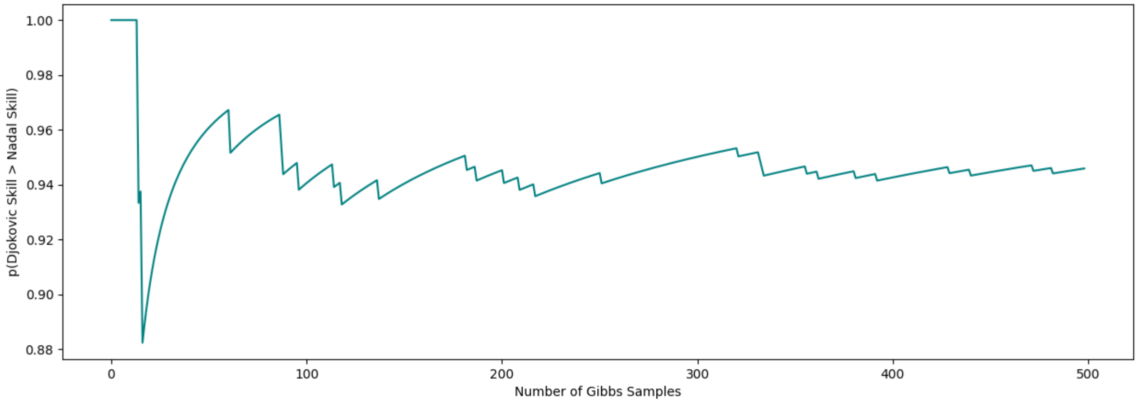

Figure 9: Probability of Djokovic being more skillful than Nadal by empirically comparing the Gibbs samples, as a function of the number of samples used.

From Figures 7 and 8, the data seems to be approximately Gaussian, which is why the first two methods provide good results. Given that method 2 requires less samples than method 3 and does not ignore the covariance between skill, this method will be used to compute the probabilities for the entire top four. The probabilities are given in Table 3 and are quite similar to those calculated by EP in Table 1 (left). However, EP was much faster.

	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.959	0.937	0.987
Nadal	0.041	-	0.415	0.764
Federer	0.063	0.585	-	0.804
Murray	0.013	0.236	0.196	-

Table 3: Probability of row player being more skillful than column player, calculated by approximating joint Gaussian distributions between each pair of players from the Gibbs samples, and sampling random variables from each joint distribution.

Listing 4: Code for calculating probability of Djokovic being more skillful than Nadal by approximating marginal skills as Gaussians.

```
for i in range(2): #Iterate over two players
    data.append(thinned_samples[top_four[i]]) #For method 2
    means[i]=np.mean(thinned_samples[top_four[i]]) #Mean
    stds[i]=np.std(thinned_samples[top_four[i]]) #Standard deviation
mean = means[1]-means[0]
var = stds[0]**2 + stds[1]**2
prob = 1 - norm.cdf(mean/var**0.5)
```

Listing 5: Code for calculating probability of Djokovic being more skillful than Nadal by approximating joint skill as multivariate Gaussian.

```
cov = np.cov(data[0], data[1]) #Covariance matrix
dist = multivariate_normal(means, cov) #Multivariate Gaussian
it = 50000 #Number of random variables to sample from distribution
count = 0
for _ in range(it):
    sample = dist.rvs() #Sample pair of random variables
    if sample[0] > sample[1]: #If Djokovic > Nadal
        count += 1
prob = count/it
```

Listing 6: Code for calculating probability of Djokovic being more skillful than Nadal directly from the Gibbs samples.

```python
count = 0
for i in range(len(data[0])):
    if data[0][i] > data[1][i]:
        count += 1
prob = count/len(data[0])
```

5 Task E

The rankings of players based on three different methods:

1. empirical win average,
2. Gibbs sampling for True-Skill, and
3. EP for True-Skill

are presented in Figure 10, with players sorted by Gibbs rank.

The empirical rankings are ineffective as opponent skill level is not considered - some players with zero win rates have higher Gibbs and EP rankings than those with non-zero win rates (but with larger uncertainties, as shown in Figure 11), as they have faced tougher opponents. Empirical win averages also give the same ranking to players with the same win ratio (regardless of opponent difficulty). Gibbs and EP give very similar rankings, with small differences due to approximations in EP and a finite number of samples in Gibbs. The MP algorithm converges quicker than Gibbs, but Gibbs converges to the true joint skill distribution whilst EP converges to approximate marginals. Therefore, the Gibbs rankings should be more accurate, hence the players are sorted by Gibbs ranking in Figures 10 and 11. Figure 12 shows the player rankings based on predicted game outcomes rather than skill by computing the probabilities of beating every other player from the Gibbs and EP skills, and then averaging - the ordering of players is almost identical.

Listing 7: Code to find probability of winning for every player against every other player, based on Gibbs and EP, and then average probabilities over all opponents.

```python
ep_probs = np.zeros(shape=(M,M))
gibbs_probs = np.zeros((M, M))
for i in range(M):
    for j in range(M):
        gibbs_probs[j, i] = np.mean(norm.cdf(thinned_samples[i] - thinned_samples[j]))
        mean = mean_player_skills[i] - mean_player_skills[j]
        variance = (1 / precision_player_skills[i]) + (1 / precision_player_skills[j]) + 1
        p_win = norm.cdf(mean/ np.sqrt(variance))
        ep_probs[i,j] = p_win

ep_winning_probs = np.zeros(M)
gibbs_winning_probs = np.zeros(M)
for p in range(M):
    gibbs_winning_probs[p] = np.sum(gibbs_probs[:, p]) / (M - 1)
    ep_winning_probs[p] = np.mean(ep_probs[p,:])
```

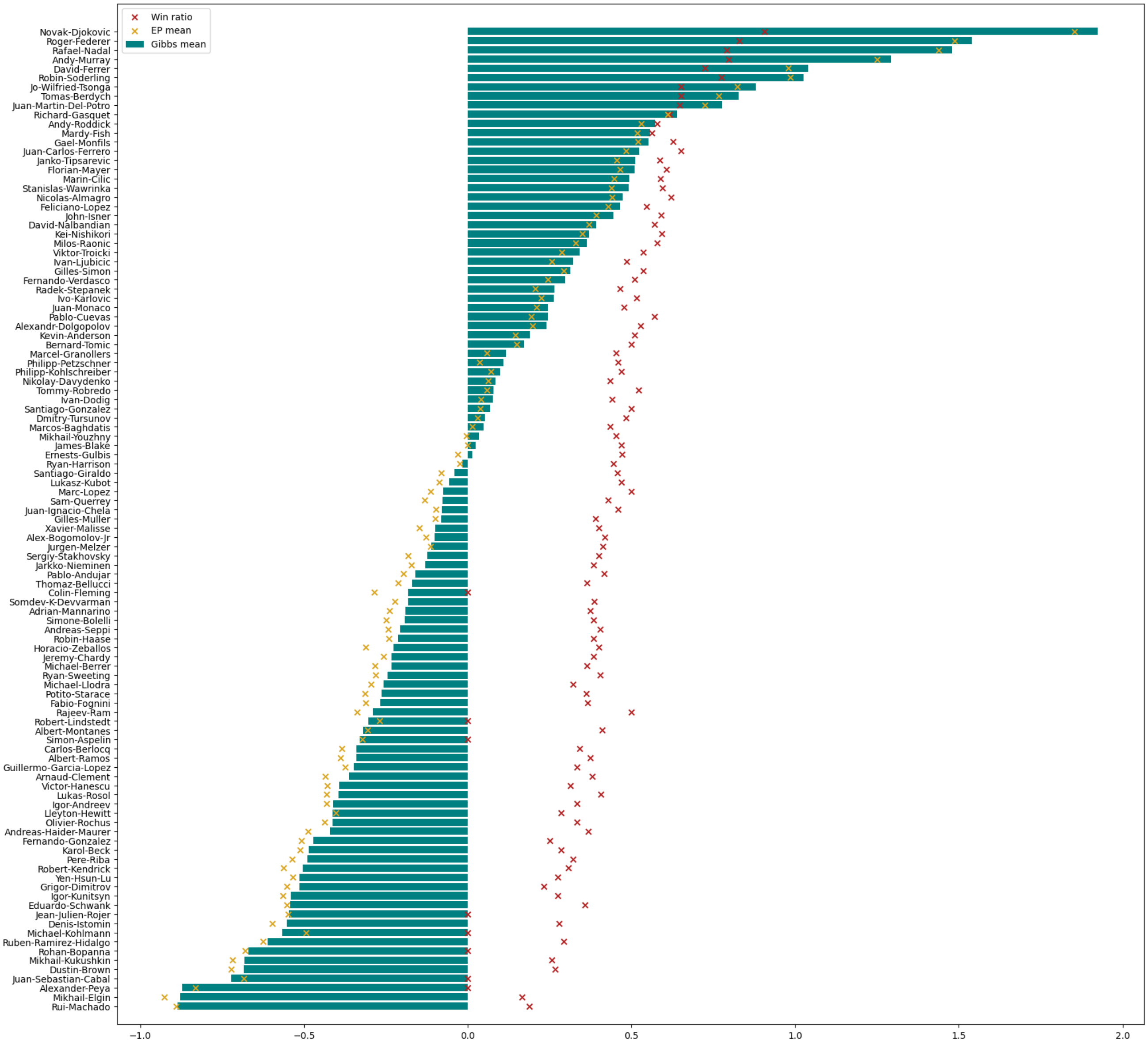

Figure 10: Rankings of players using the three different methods, and sorted by Gibbs sample mean.

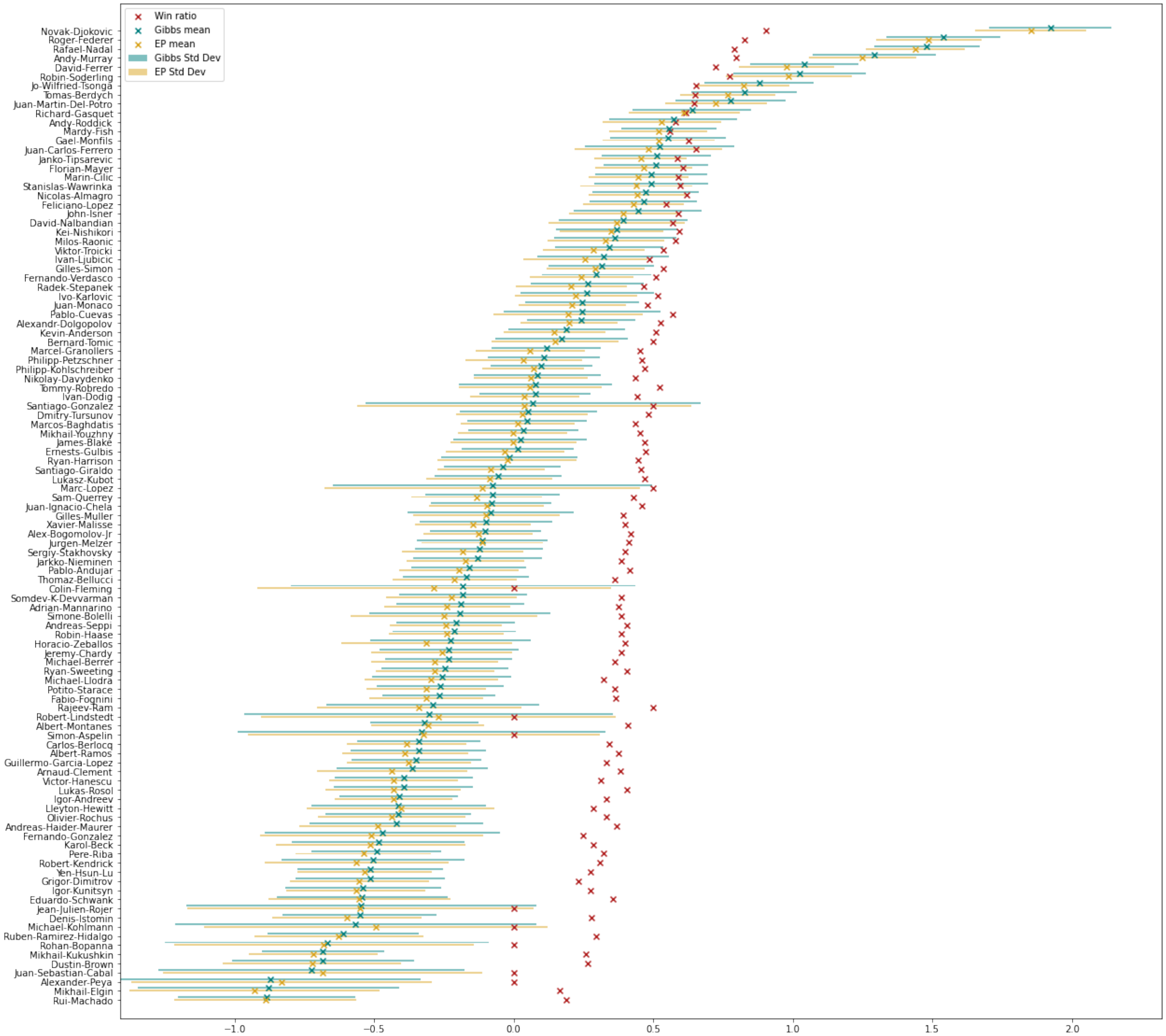

Figure 11: Rankings of players using the three different methods, and sorted by Gibbs sample mean, with standard deviations

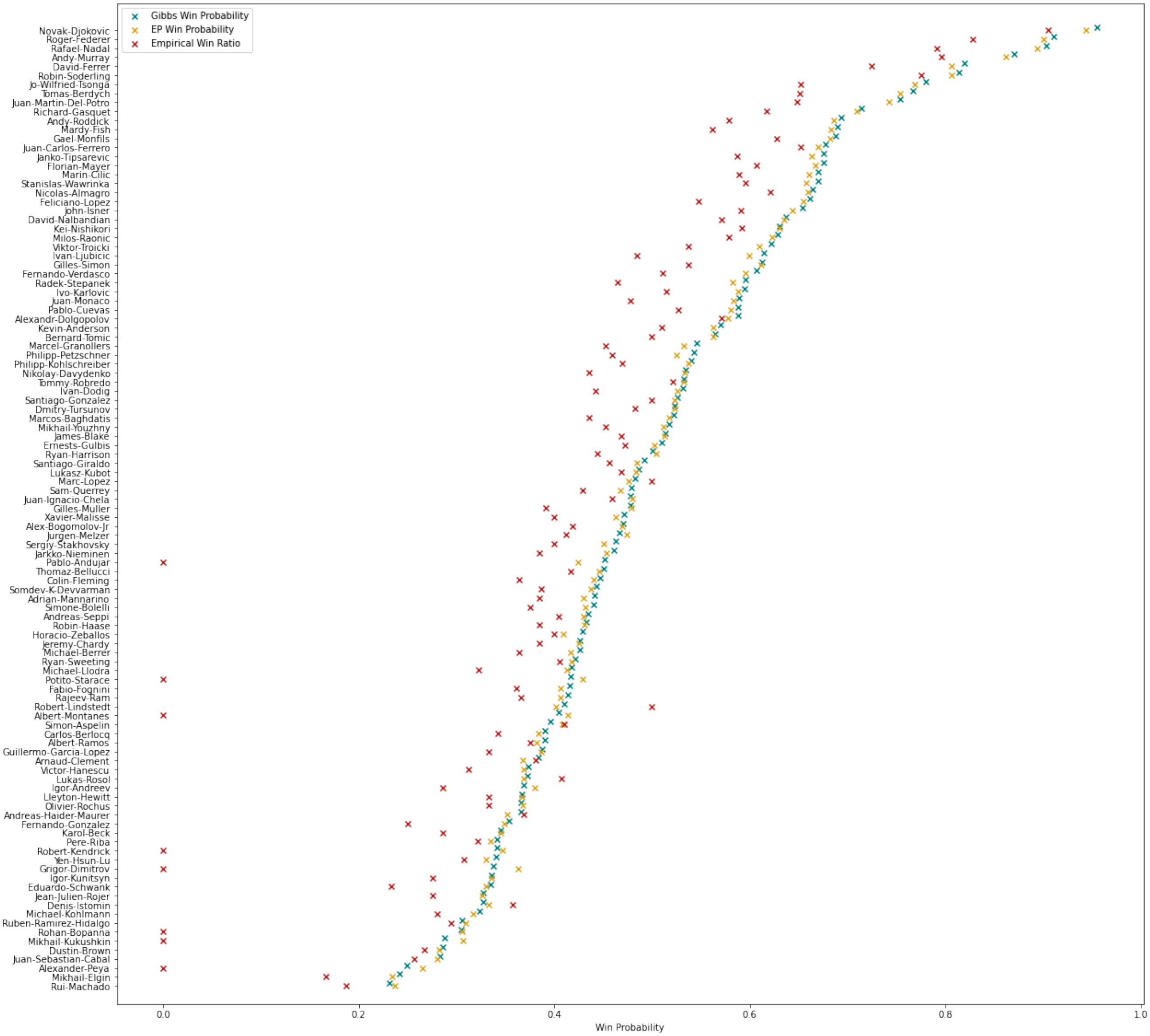

Figure 12: Rankings of players based on win probability, using three different methods, and sorted by Gibbs win probability.

References

[1] John F. Geweke. Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments. Technical report, 1991.