CUED - Engineering Tripos Part IIB 2024-2025 Module Coursework

Module	4F13	Title of report	Probabilistic Ranking

Date submitted: 22/11/24	Assessment for this module is ☑ 100% / □ 25% coursework of which this assignment forms 33 %

UNDERGRADUATE and POST GRADUATE STUDENTS		
Candidate number:	5488A	☑ Undergraduate □ Post graduate

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:			

Marker: Date:

4F13 Coursework 2 - Probabilistic Ranking

November 2024

1 Task A

```
for p in range(M):
    m[p] = np.dot(t[:,0], 1*(G[:,0] == p)-1*(G[:,1] == p))
for g in range(N):
    w = G[g, 0]; l = G[g, 1]
    iS[w, w] += 1; iS[w, l] -= 1; iS[l, w] -= 1; iS[l, l] += 1
```

Listing 1: Python calculation of mean and covariance matrix of skills given performances for Gibbs sampling.

Figure 1 shows samples of the skills at each Gibbs iteration for 3 players. We see that at each iteration the sampled skill of each player appears random but perhaps not independent of the previous iteration. This is expected from an MCMC method like Gibbs sampling and shown better by the auto covariance plot in Figure 2.

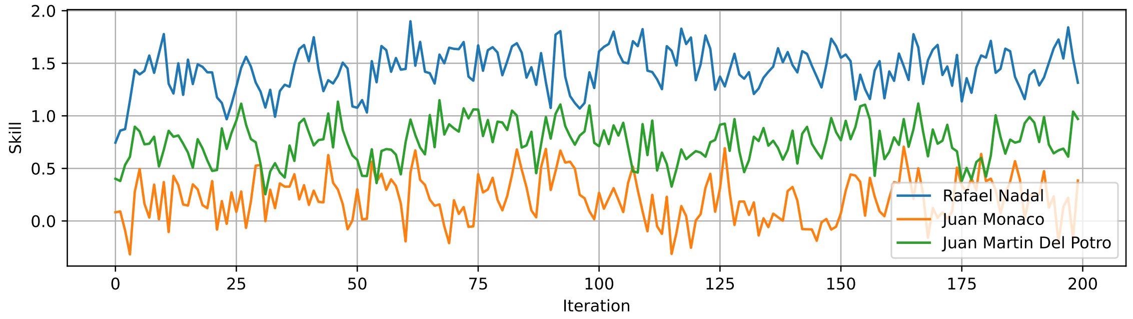

Figure 1: Samples of the skill of 3 players at each Gibbs iteration.

If the samples at each iteration were independent we would expect the auto covariance to be zero for all non zero lags. However, we notice that the auto covariance only converges to zero for all players for lags greater than 10. This indicates that if we kept only every 10th sample they would be independent, this is the logic behind thinning. We do not need to employ thinning techniques here as we are not running into memory issues but it is a useful tool for large datasets.

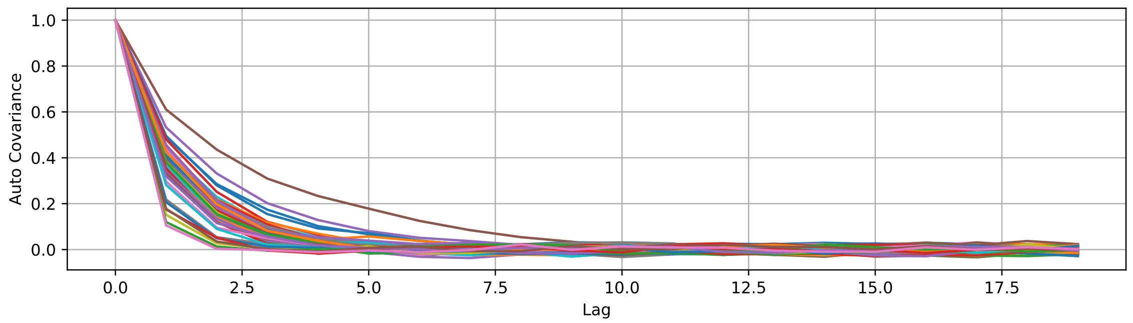

Figure 2: Skill sample auto covariance for all players

The convergence of the skill population is shown in Figure 3. We see that there is no clear burn in time for the population mean but the standard deviation takes about 5 iterations to enter the high probability region. This makes sense from our auto covariance analysis as we concluded samples 10 iterations apart are roughly independent so the burn in time should be no more than 10 iterations. For any further analysis we will discard the first 100 Gibbs iterations as burn in time to be sure.

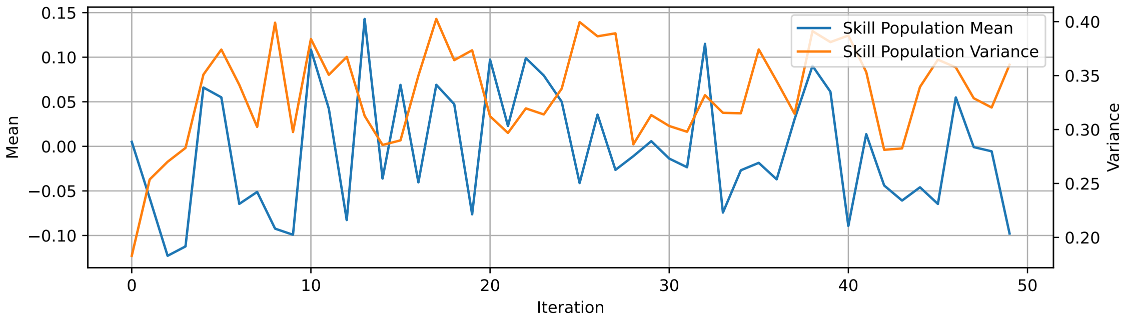

Figure 3: Skill populations mean and standard deviation with Gibbs iteration.

To estimate how many iterations we need to run the Gibbs sampler to get skill estimates within a certain tolerance we can analyse the variance of our Monte Carlo skill estimates. Equation 1 calculates the variance of the Gibbs skill estimate, $\bar{w_i}$, for player i after N iterations. The variance depends on the sum of the auto covariance, $\frac{\sigma_i^{(t)2}}{\text{Var}(w_i)}$, and is inversely proportional to N. We compute the average of this variance increase term over all players from the data in Figure 2 and find a variance increase factor of 4. From these calculation we find that in order to achieve a skill estimate variance 10^{-3} smaller than our posterior skill variance we must run our Gibbs sampler for 4000 iterations.

$$\begin{aligned}\text{Var}(\bar{w_i}) = \frac{1}{N}\text{Var}(\sum_{n=1}^{N} w_i^{(n)}) = \frac{\text{Var}(w_i)}{N}(1 + 2\sum_{t=1}^{N-1}(1 - \frac{t}{N})\frac{\sigma_i^{(t)2}}{\text{Var}(w_i)}) \\ \approx \frac{\text{Var}(w_i)}{N}(1 + 2\sum_{t=1}^{\infty}\frac{\sigma_i^{(t)2}}{\text{Var}(w_i)}) \approx 4\frac{\text{Var}(w_i)}{N}\end{aligned} \tag{1}$$

Convergence is not the only factor in considering the computational complexity of the Gibbs sampler, we must also consider the complexity of each iteration. There are three main computations each iteration, the mean vector, the covariance matrix and its Cholesky decomposition. If we denote the number of players as M and the number of games as G then the total complexity is $\mathcal{O}(M^3 + MG)$ with the following breakdown.

- Mean vector $\mathcal{O}(MG)$: M dot products of length G.
- Covariance matrix $\mathcal{O}(G)$: G iterations to populate the matrix.
- Cholesky decomposition $\mathcal{O}(M^3)$: Covariance matrix is $M \times M$.

2 Task B

In the Gibbs sampling approach we are sampling from a Markov chain with stationary distribution $p(\mathbf{w}|\mathbf{y})$ and we wish to compute expectations of the form $\mathbb{E}_{\mathbf{w}\sim p(\mathbf{w}|\mathbf{y})}[f(\mathbf{w})]$. The distribution of our samples must first converge to the stationary distribution, the burn-in period, and then our expectation estimate needs to converge too. From the previous section we know that the variance of the expectation is proportional to N^{-1}.

With message passing we are converging to a stable graph with the distribution of variables at the nodes being approximated by Gaussians. We can say that the graph has converged within a tolerance ϵ at iteration n when the absolute change of all marginal parameters between iterations are less than ϵ. Equation 2 formalises this with $m_i^{(n)}$ and $p_i^{(n)}$ being the estimated mean and precision of player i at iteration n.

$$|m_j^{(n)} - m_i^{(n-1)}| < \epsilon \quad \text{and} \quad \max_i |p_i^{(n)} - p_i^{(n-1)}| < \epsilon \quad \forall i \tag{2}$$

We plot the largest absolute change between iterations for each parameter in Figure 4. By inspection of Figure 4 we notice that convergence is exponential and we achieve a tolerance of machine precision within 300 iterations. This is much faster than the Gibbs sampler where $\epsilon \propto \sqrt{\text{Var}(\bar{w_i})} \propto N^{-0.5}$.

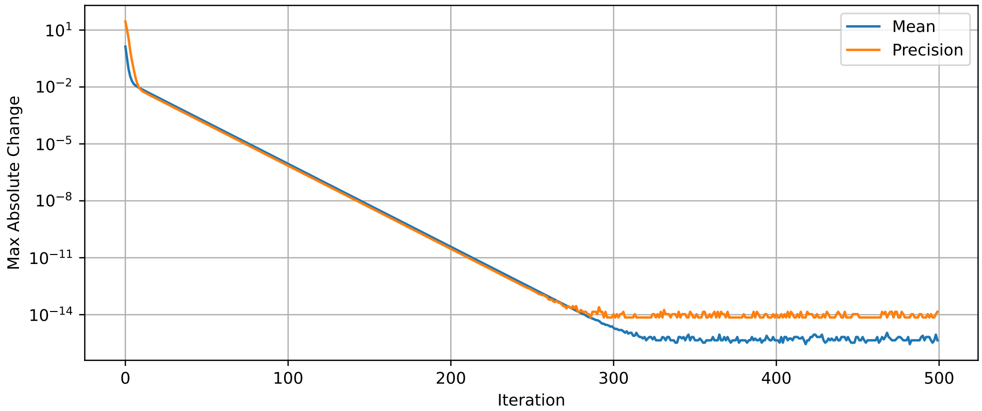

Figure 4: Convergence of the mean and precision of the marginal skill of all players.

$P(s_{ij} > 0)$	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.94	0.91	0.99
Nadal	0.06	-	0.43	0.77
Federer	0.09	0.57	-	0.81
Murray	0.01	0.23	0.19	-

(a) Probability that row player is more skilled that column player.

$P(t_{ij} > 0)$	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.66	0.64	0.72
Nadal	0.34	-	0.48	0.57
Federer	0.36	0.52	-	0.59
Murray	0.28	0.43	0.41	-

(b) Probability that row player wins a match against column player.

Table 1: Skill and performance differences between top 4 ATP players based on EP.

3 Task C

Using the skill parameters estimated by the EP algorithm we denote the skill of player i as $w_i \sim \mathcal{N}(m_i, p_i^{-1})$, with m_i and p_i being the mean and precision for player i. The distributions of the skill difference, $s_{ij} = w_i - w_j$, and performance difference, $t_{ij} = s_{ij} + \eta : \eta \sim \mathcal{N}(0, 1)$, between player i and j are given in Equation 3. These are simple to compute under the assumption that w_i, w_j and η are independent.

$$\begin{aligned} s_{ij} &\sim \mathcal{N}(m_i - m_j, p_i^{-1} + p_j^{-1}) \\ t_{ij} &\sim \mathcal{N}(m_i - m_j, p_i^{-1} + p_j^{-1} + 1) \end{aligned} \tag{3}$$

We wish to compute the probability that player i has greater skill that player j, $P(s_{ij} > 0)$, and the probability that player i will win a match against player j, $P(t_{ij} > 0)$. We use the identity $P(x > 0) = \Phi(\frac{\mu}{\sigma})$ where Φ is the standard normal c.d.f. and $x \sim \mathcal{N}(\mu, \sigma^2)$. We find these probabilities between the top 4 ranked ATP players in Table 1.

We are always more confident in who is more skilled that who will win a match. This makes sense as $\sigma_t > \sigma_s$, however, intuitively this is also clear as we are always less certain about the outcome of a match due to performance variation.

4 Task D

Using our Gibbs samples we would like to estimate the skill difference between Nadal and Federer. We investigate three methods to do this in the following sections.

4.1 Approximate marginal skills with Gaussians

We compute the sample marginal mean and variance for each player and model their skills as $w_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. Figure 5 shows the marginal histograms and Gaussian approximation, Table 2 contains the parameters for the approximation. Using the formulas from Task C we find that $P(w_{Nadal} > w_{Federer}) = 0.438$. Federer has the higher skill mean with a higher variance than Nadal.

μ_{Nadal}	$\mu_{Federer}$	σ^2_{Nadal}	$\sigma^2_{Federer}$	$\sigma^2_{Nadal,Federer}$
1.48	1.52	0.038	0.041	0.009

Table 2: Parameters for Gaussian fit of marginal and joint skill Gibbs samples for Nadal and Federer

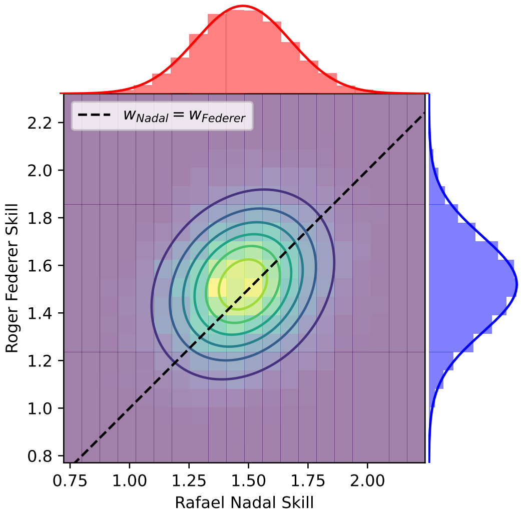

Figure 5: Joint and marginal skill distribution (histograms), joint Gaussian approximation (contours) and marginal gaussian approximation (sides), for Nadal and Federer.

4.2 Approximate joint skill distribution with a multivariate Gaussian

We can also approximate the joint skill distribution with a Gaussian by computing the sample covariance matrix, these parameters are again given in Table 2. We find a positive covariance between the two players, $\sigma^2_{Nadal,Federer} = 0.009$. We visualise the joint distribution and our approximation in Figure 5, it is clear that the covariance skews the join distribution enough that the marginal approximation is not accurate. We find $P(s_{ij} > 0) = 0.430$.

4.3 Use the samples directly

We can also compute a Monte Carlo estimate of $P(w_{Nadal} > w_{Federer})$. This estimate is given in Equation 4 where $(w_i^{(n)}, w_j^{(n)})$ are samples from a Markov chain with stationary distribution $p(w_i, w_j)$. Our Gibbs samples satisfy this condition and we find $P(w_{Nadal} > w_{Federer}) = 0.431$.

$$\mathrm{P}(w_i > w_j) = \int \mathbb{1}(w_i > w_j) p(w_i, w_j) dw_i dw_j \approx \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}(w_i^{(n)} > w_j^{(n)}) \tag{4}$$

4.4 Comparison

The skill posterior is not necessarily Gaussian, therefore, computing $\mathrm{P}(w_i > w_j)$ with the Monte Carlo method is the only unbiased method. However, as observed the Gaussian approximation is close. Estimating $\mathrm{P}(w_i > w_j)$ with the marginal approximation has $\mathcal{O}(M)$

$P(s_{ij} > 0)$	Djokovic	Nadal	Federer	Murray
Djokovic	-	0.95	0.92	0.99
Nadal	0.05	-	0.43	0.78
Federer	0.08	0.57	-	0.81
Murray	0.01	0.22	0.19	-

Table 3: Probability that row player is more skilled that column player for top 4 ATP players using Gibbs samples

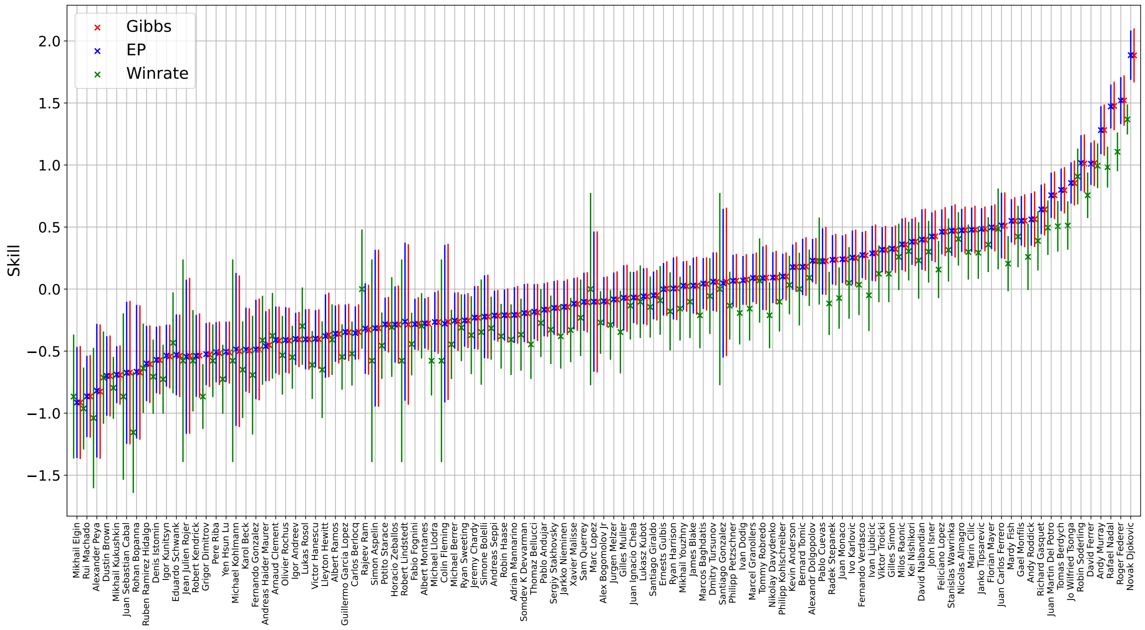

Figure 6: Posterior skill mean and one standard deviation error bars for all players from each model.

complexity compared to $\mathcal{O}(M^2)$ for the joint approximation and direct method where we must compute sums for all pairs of players. Therefore, when comparing a large number of players the small decrease in accuracy of the marginal approximation may be acceptable. As we are only comparing 4 players we will use the direct method for its superior accuracy.

We notice that the probabilities in Table 3 are very similar to those in Table 1a. This makes sense as the message passing algorithm approximates the posterior as a Gaussian and the Gibbs samples show this is accurate. The message passing EP algorithm achieves the same performance as marginal Gibbs method but with much faster convergence.

5 Task E

We have explored methods for skill estimation using both Gibbs sampling and EP algorithms. A simpler model for estimating skill is to assume every player has a fixed win rate r_i. The posterior distribution of r_i for a player, i, who wins k_i out of n_i matches with a uniform prior, $r_i \sim \mathrm{U}(0,1)$, is the Beta distribution given in Equation 5. Also given in Equation 5 are the formulas for the posterior mean and variance. Skill, in this model, is a linear transformation of win rate where $w_i = \sqrt{12}(r_i - 0.5)$. This transformation means the skill prior has zero mean and unit variance for easier model comparison.

$$\begin{gathered} r_i | k_k, n_i \sim \mathrm{Beta}(1 + k_i, 1 + n_i - k_i) \\ \mathrm{E}(r_i | k_i, n_i) = \frac{1 + k_i}{2 + n_i} \quad \mathrm{Var}(r_i | k_i, n_i) = \frac{(1 + k_i)(1 + n_i - k_i)}{(2 + n_i)^2 (3 + n_i)} \end{gathered} \tag{5}$$

We show the posterior skill mean and one standard deviation error bars for all players from each model in Figure 6. This again shows, as discussed at the end of Task D, that the message passing model achieves similar results to the Gibbs sampler with much faster convergence.

The win rate model ranks the players similarly to the other models while having linear time complexity in the number of players. However, it has a major weakness, the skill of the opponents in each game does not factor into the skill calculation. The model could therefore be exploited by playing many games against lower ranked players to boost ones ranking. Andy Murray is an example of a top player who is ranked differently by the win rate model, being ranked above Roger Federer, because although he has a higher win rate, on average his matches have been against lower ranked players.